



**UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA**

FABRICIO SOARES DA SILVA

**Uma Extensão do Diagrama de Classe em
Aspect-Oriented Design Model (AODM) para modelagem
de *Design by Contract***

Recife

2017

FABRICIO SOARES DA SILVA

**Uma Extensão do Diagrama de Classe em
Aspect-Oriented Design Model (AODM) para modelagem
de *Design by Contract***

Este trabalho foi apresentado à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre Profissional em Ciência da Computação.

Área de Concentração: **Sistemas de Informação**

Orientador: **Prof. Henrique Emanuel Mostaert Rebêlo**

Recife
2017

Catálogo na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

S586e Silva, Fabrício Soares da
Uma extensão do diagrama de classe em *aspect-oriented design model* (AODM) para modelagem de *design by contract* / Fabrício Soares da Silva. – 2017.
132 f.: il., fig., tab.

Orientador: Henrique Emanuel Mostaert Rebêlo.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2017.
Inclui referências, apêndices e anexos.

1. Engenharia de software. 2. AODM. I. Rebêlo, Henrique Emanuel Mostaert (orientador). II. Título.

005.1

CDD (23. ed.)

UFPE- MEI 2018-127

FABRICIO SOARES DA SILVA

**Uma Extensão do Diagrama de Classe em
Aspect-Oriented Design Model (AODM) para modelagem
de *Design by Contract***

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre Profissional em 28 de dezembro de 2017.

Aprovado em: 28/12/2017

Prof. Leopoldo Motta Teixeira
Centro de Informática / UFPE

Prof. Gilberto Amado de Azevedo Cysneiros Filho
Universidade Federal Rural de Pernambuco

Prof. Henrique Emanuel Mostaert Rebêlo
Centro de Informática / UFPE
(Orientador)

Este trabalho é dedicado principalmente a DEUS e aos meus familiares e amigos que são os Aspectos mais importantes do meu código.

AGRADECIMENTOS

Agradeço primeiramente a Deus que me sustenta e me ampara nos momentos mais difíceis e me dá forças para superar desafios como o de concluir um Mestrado. Agradeço também por me oferecer sempre, após uma semana intensa de estudos e trabalho em um Estado, que apesar de não ser o meu, me acolheu durante esse par de anos, todos os pores do sol que sempre do lado direito do avião, em meus regressos ao Rio de Janeiro recarregavam minha bateria com sua energia impar, sua luz e sua beleza alaranjada e cintilante ao mergulhar de corpo e alma sob as nuvens.

Agradeço aos meus pais, José Carlos e Suzete, que sempre me deram força e acreditaram em mim, não importa a besteira que eu fizesse. Porém, sempre com retidão e carinho, acima de tudo, me reconduziam aos trilhos quando essa besteira era grande demais. Agradeço também a minha tia Cizete que, junto com meus pais, me incentivaram e investiram em minha formação de caráter e acadêmica desde o berço.

Agradeço a paciência, dedicação e amor da minha esposa Elaine que foi minha maior incentivadora para encarar esse desafio, suportando sem reclamar todas as minhas ausências. E as minhas filhas Luísa e Clarissa, que com a sabedoria precisa, peculiar das crianças, souberam fazer suas artimanhas, peraltices e travessuras de modo a me fazer levantar da cadeira nos momentos mais delicados de meu estudo, fato que me tirava do sério momentaneamente, mas logo em seguida percebia que os seus sorrisos me aliviavam o estresse e enfim, regressava melhor para o meu trabalho.

Agradeço a minha tia Mara, meu irmão Rodrigo, minha cunhada Maria Gerla e meus sobrinhos Davi e Gustavo, pela família maravilhosa que me proporcionam e pela presença em cada momento vivido. Agradeço também a Sônia que com sua dedicação incansável ajudou a conter os dois furacõezinhos que tenho em casa, durante todas as minhas ausências.

Agradeço a pessoas especiais que passaram por minha vida e hoje estão no céu, como anjos, a interceder por minhas vitórias, como meus avós (Odette - "a senhora sempre esteve presente comigo-, Cezino, Alair e José) e minha filhinha Clara, meu eterno anjinho.

Agradeço de coração a todos os meus professores, sem exceção, desde o jardim da infância até o mestrado. Sem vocês eu não seria nada. Vocês são meus Mestres.

Agradeço ao Clube de Regatas do Flamengo por todas as alegrias proporcionadas desde sempre.

Agradeço aos meus colegas de mestrado, que de cada canto do Brasil, enchem o auditório da UFPE com entusiasmo e perseverança de construirmos um futuro melhor para

a educação do nosso país e prover tecnologia para resolver todos os problemas do universo. Pra vocês os meus mais sinceros: "Gil, Gil, Gil, Gil, Gil..." Vocês são inesquecíveis. Dentre tantos colegas, queria agradecer especialmente ao Andrei, Gil Cristiano, Wilton e José Henrique, que foram companheiros para todas as horas. "Manda um abraço pra sua mãe, Zé". Que o Pará, Bahia, Piauí e Pernambuco possam aproveitar os grandes profissionais e amigos que vocês são. Também agradeço o companheirismo do Rogério, Tiago Heineck, Plínio e Alan. Bem como os colegas dos outros cursos como o David da Turma de Redes. Vocês todos são nota 1010.

Agradeço aos cafés da manhã e ao teto da Pousada Pinheiros, que foi um QG nessa etapa de formação. Ao Boi Brasa, aos crepes, as pizzas que me sustentaram nessa jornada. À Praia de Boa Viagem cujos ventos varreram minhas frustrações, e me renovaram com ar novo me lembrando sempre que como na vida, o horizonte é lindo, mas há tubarões.

Agradeço também aos meus companheiros e amigos do Colégio Pedro II que me deram todo o suporte que eu precisava nos momentos mais complicados. Obrigado, Prof. Vera Medalha, Prof. Luiz Francisco, Prof. Jorge Fernando, Márcio Doval, Thiago Martoreli, Soraia Felício, a todos os meus colegas da DTI, em especial meus companheiros da SPSPN. Sem vocês nada disso seria possível.

Agradeço a SETEC/MEC e a UFPE pela iniciativa de se criar esta turma de Mestrado Profissional, dando oportunidade de formação acadêmica especializada a tantos profissionais da educação federal deste país.

Agradeço o carinho das servidoras da secretaria da Pós-Graduação Leila Oliveira de Azevedo e Silva e Joelma Souza de Menezes França que com dedicação e profissionalismo zelaram por nós durante todo o curso com uma competência incomparável, enchendo de orgulho o serviço público.

Agradeço especialmente ao Prof. Henrique Emanuel Mostaert Rebêlo, e a todos os professores da UFPE que me ajudaram em minha trajetória, em particular os professores Fernando Castor, Leopoldo, André, Carla, Giordano, Carlos e Jaelson, por toda a inspiração de saber que inflaram meus pulmões.

Agradeço a cada amigo, parente, colega ou conhecido, que me fortaleciam a cada gesto ou mesmo na ausência e na distância desses momentos de dedicação, meu coração recebeu a energia de suas torcidas, porém a limitação das laudas me impedem de citá-los um a um.

Por fim, agradeço a poesia e ao código, que na expressão do meu traçar, equilibram a essência da minha razão e da minha emoção.

Termino o livro e fecho o computador sabendo que por mais que os escritores escrevam, os músicos componham e cantem, os pintores e escultores joguem com formas, cores e luzes -, por mais que o contexto paralelo da arte expresse o profundo contraditório sentimento humano, embora dance à nossa frente e nos convoque até o último fio de lucidez, o essencial não tem nome nem forma: é descoberta e assombro, glória ou danação de cada um. (LUFT, 2003, p. 156)

RESUMO

Com a evolução da técnica de desenvolvimento de *Design by Contract* (DbC), muitas implementações vem surgindo e em diversas linguagens de programação. Os contratos são elementos importantes para a validade dos métodos e com a implementação de DbC em Aspectos, os contratos passam a ser elementos importantes também para os *advice*. Com isso, o registro e a visualização dos contratos, tanto as premissas quanto as validações de saída, em fase de análise de projeto ganham uma devida importância. Porém, com as técnicas existentes para se diagramar classes e aspectos, esta não é uma atividade possível. Nesta pesquisa está descrita uma extensão do artefato de "Diagrama de Classes" com suporte a Aspectos provido pela técnica AODM, com o objetivo específico de expressar contratos estabelecidos na técnica *Design by Contract* em classes e aspectos, sem ferir os princípios da UML. Para auxiliar no desenvolvimento da pesquisa e criar exemplos que pudessem ser modelados com a técnica desenvolvida nesta dissertação, optou-se pelo uso das sintaxes AspectJ/@AspectJ (para aspectos), JML (para DbC) e AspectJML (para DbC sobre Aspectos). Como resultado foi desenvolvido uma técnica que estende a AODM para se modelar contratos tanto em classes quanto em aspectos, se utilizando de artifícios já suportados pela UML através de estereótipos e propriedades. Esta técnica proporciona que todos estes elementos possam ser representados em apenas um único modelo visando oferecer uma visão ampla do sistema que está sendo desenvolvido.

Palavras-chave: *Aspect Oriented Modeling*. AspectJ. AspectJML. *Design by Contract*. JML. UML.

ABSTRACT

With the evolution of the design technique of Design by Contract (DbC), many implementations have been emerging and in several programming languages. Contracts are important elements for the validity of the methods and with the implementation of DbC in Aspects, contracts become important elements for advice as well. With this, the registration and the visualization of the contracts, both the assumptions and the exit validations, in the phase of project analysis, are given due importance. However, with the existing techniques to diagram classes and aspects, this is not a possible activity. In this research, an extension of the "Class Diagram" artifact with support to Aspects provided by the AODM technique is described, with the specific objective of expressing contracts established in the Design by Contract technique in classes and aspects, without hurting the UML principles. AspectJ/@AspectJ (for aspects), JML (for DbC) and AspectJML (for DbC over Aspects) were used to support the development of the research and create examples that could be modeled using the technique developed in this dissertation. As a result, a technique has been developed that extends AODM to model contracts in both classes and aspects, if using artifacts already supported by UML through stereotypes and properties. This technique provides that all these elements can be represented in a single model to offer a broad view of the system that is being developed.

Keywords: Aspect Oriented Modeling. AspectJ. AspectJML. Design by Contract. JML. UML.

LISTA DE FIGURAS

Figura 1 – Exemplo de Classes com <i>Crosscutting Concerns</i>	18
Figura 2 – Exemplo Aspectos	18
Figura 3 – Exemplo Classe	23
Figura 4 – Método para condução da <i>Design Science Research</i>	32
Figura 5 – Método para condução da Revisão Sistemática da Literatura	36
Figura 6 – Resumo da Revisão Sistemática da Literatura	49
Figura 7 – Representação de um Aspecto conforme AODM	60
Figura 8 – Representação de um <i>Join Point</i> conforme AODM	60
Figura 9 – Exemplo de Modelo AODM	61
Figura 10 – Exemplo de AODM	69
Figura 11 – Modelo AODM Estendido	76

LISTA DE GRÁFICOS

Gráfico 1 – Publicações em Fontes de Dados x Horizonte Temporal	43
Gráfico 2 – Classificação Inicial das Publicações	46
Gráfico 3 – Classificação das Publicações do <i>Snowballing</i>	47
Gráfico 4 – Classificação Final das Publicações	48
Gráfico 5 – Publicações x Horizonte Temporal	50
Gráfico 6 – Categorização das Publicações	51

LISTA DE QUADROS

Quadro 1 – <i>Frameworks</i> POA	19
Quadro 2 – Tipos de Artefatos da <i>Design Science Research</i>	29
Quadro 3 – Critérios para a condução das pesquisas que utilizam a <i>Design Science Research</i>	30
Quadro 4 – Etapas da <i>Design Science Research</i> x Estrutura de Capítulos da Dissertação	33
Quadro 5 – Etapas e Atividades da Revisão Sistemática da Literatura	35
Quadro 6 – Síntese dos Dados: Aspectos	53
Quadro 7 – Síntese dos Dados: Componentes do Aspecto	54
Quadro 8 – Matriz da Síntese dos Dados sobre AOM	55
Quadro 9 – Matriz da Síntese dos Dados: <i>Design by Contract</i> sobre AOP	56
Quadro 10 – Resumo das várias apresentações de atributos explícitos no UML	64
Quadro 11 – Dicionário de Dados dos Estudos	92
Quadro 12 – Listagem das Publicações Seleccionadas pela RSL	94

LISTA DE TABELAS

Tabela 1 – Publicações em Fontes de Dados x Horizonte Temporal	44
Tabela 2 – Descarte de Publicações Duplicadas	45
Tabela 3 – Descarte de Publicações Após Avaliação de Títulos	45
Tabela 4 – Descarte de Publicações Após Avaliação de Resumos e Palavras-Chave	45
Tabela 5 – Descarte de Publicações Após Leitura Completa do Trabalho	45
Tabela 6 – Classificação Inicial das Publicações	46
Tabela 7 – <i>Snowballing</i> : Avaliação das Publicações	47
Tabela 8 – Classificação das Publicações do <i>Snowballing</i>	47
Tabela 9 – Classificação Final das Publicações	48
Tabela 10 – Categorização das Publicações	51
Tabela 11 – Relação dos Meios de Publicações dos Periódicos	123

LISTA DE ABREVIATURAS E SIGLAS

AO-UML	<i>Aspect Oriented - Unified Model Language</i> , ou "Orientação a Aspecto - Linguagem de Modelagem Unificada"
AODM	<i>Aspect-Oriented Design Model</i> , ou "Modelo de Projeto Orientado à Aspecto"
AOM	<i>Aspect Oriented Modeling</i> , ou "Modelo Orientado à Aspecto"
BDTD	Biblioteca Digital Brasileira de Teses e Dissertações
CASE	<i>Computer-Aided Software Engineering</i> , ou "Engenharia de Software Assistida por Computador"
CESM	<i>Composition, Environment, Structure, Mechanism</i> , ou "Composição, Ambiente, Estrutura, Mecanismo"
DbC	<i>Design by Contract</i> , ou "Programação por Contrato"
DS	<i>Design Science</i> , ou "Ciência do Artificial"
DSR	<i>Design Science Research</i> , ou "Pesquisa da Ciência do Artificial"
IBICT	Instituto Brasileiro de Informação em Ciência e Tecnologia
JDK	<i>Java Development Kit</i> , ou Pacote para Desenvolvimento Java
JML	<i>Java Modeling Language</i> , ou "Linguagem de Modelagem para Java"
JRE	<i>Java Runtime Environment</i> , ou Ambiente de Tempo de Execução Java
JVM	<i>Java Virtual Machine</i> , ou Máquina Virtual Java
OMG	<i>Object Management Group</i> , ou "Grupo de Gerenciamento de Objetos"
OO	Orientação a Objetos
POA	Programação Orientada a Aspectos
RSL	Revisão Sistemática da Literatura
UML	<i>Unified Modeling Language</i> , ou "Linguagem de Modelagem Unificada"
UMLAUT	<i>UML All pUrpose Transformer</i> , ou "Transformador UML multifuncional"
UXF/a	<i>UML eXchange Format, aspect extension</i> , ou "Formato de intercâmbio em UML com extensão em aspectos"

SUMÁRIO

1	INTRODUÇÃO	17
1.1	Justificativa da Pesquisa	21
1.2	Motivação	22
1.3	Caracterização do Problema	22
1.3.1	Abordagem do Problema	23
1.4	Natureza da Pesquisa	24
1.5	Questão da Pesquisa	24
1.6	Objetivos	24
1.6.1	Objetivo Geral	24
1.6.2	Objetivo Específico	24
1.7	Organização da Dissertação	25
2	METODOLOGIA	26
2.1	Paradigma <i>Design Science</i>	26
2.2	Método de Pesquisa <i>Design Science Research</i>	28
2.3	Etapas para a condução da pesquisa	30
2.3.1	Etapa da Design Science Research x Estrutura da Dissertação	31
3	REVISÃO SISTEMÁTICA DA LITERATURA	34
3.1	Planejamento da Revisão	37
3.2	Condução da Revisão	38
3.3	Documentação da Revisão	42
3.4	Resultados da Revisão Sistemática da Literatura	42
3.4.1	Fase 1: Levantamento Inicial da Literatura	42
3.4.2	Fase 2: Avaliação da Literatura Segundo Critérios de Inclusão e Exclusão	42
3.4.3	Fase 3: Classificação das Publicações Seleccionadas	46
3.4.4	Fase 4: Busca de Literatura, Avaliação e Classificação com <i>Snowballing</i>	46
3.4.5	Fase 5: Resumo da Revisão Sistemática de Literatura	48
3.5	Síntese dos Resultados	50
3.5.1	Síntese dos Resultados de Modelagem de Aspectos	52
3.5.2	Síntese dos Resultados de <i>Design by Contract</i> sobre Aspectos	55
3.6	Critérios de Documentação da RSL	56
4	ABORDAGEM DAS TÉCNICAS BASE	57
4.1	Java	57
4.2	AspectJ	57

4.3	AODM	59
4.4	Design by Contracto	60
4.5	JML	61
4.6	AspectJML	62
5	PROJETO DO ARTEFATO	63
5.1	Desenvolvimento do Artefato	63
5.2	Exemplificação do Artefato	65
5.2.1	Exemplo (Parte 1): Código Java/AspectJ	66
5.2.2	Exemplo (Parte 2): AODM	68
5.2.3	Exemplo (Parte 3): JML	69
5.2.4	Exemplo (Parte 4): AspectJML	70
5.2.5	Exemplo (Parte 5): Extensão da AODM	72
5.3	Avaliação do Artefato	75
5.3.1	Parecer Técnico	77
6	CONSIDERAÇÕES FINAIS	78
6.1	Propostas para trabalhos futuros	79
	REFERÊNCIAS	81
	APÊNDICE A – PROTOCOLO DA PESQUISA CONFORME DESIGN SCIENCE RESEARCH	86
	APÊNDICE B – PROTOCOLO DA RSL	89
	APÊNDICE C – LISTAGEM DAS PUBLICAÇÕES SELECIONADAS PELA RSL	94
	APÊNDICE D – LISTAGEM DOS PERIÓDICOS E MEIOS DE PUBLI- CAÇÃO SELECIONADOS PELA RSL	123
	ANEXO A – PRIMEIRO PARECER TÉCNICO	127
	ANEXO B – SEGUNDO PARECER TÉCNICO	129

1 INTRODUÇÃO

A Orientação a Objetos (OO) é o paradigma (modelos de problemas e soluções) que domina o mercado de desenvolvimento de software há anos. "Considero 'paradigmas' as realizações científicas universalmente reconhecidas que, durante algum tempo, fornecem problemas e soluções modelares para uma comunidade de praticantes da ciência." (KUHN, 1998, p. 13)

A OO tem algumas limitações, tal como o tratamento de *crosscutting concerns*, ou interesses transversais¹ que são funções que se repetem em várias partes de um sistema, como por exemplo: tratamento de erros e *log*. Segundo Barreto e Lorenzi (2011), é uma técnica de registro de processos realizados de forma não volátil.

Considerando que - por meio de técnicas convencionais de modularização ou orientação de objetos - um tipo de preocupação pode ser encapsulada suavemente em blocos de construção, como módulos, classes e operações de design ou nível de código, o mesmo não é possível para outro tipo de preocupações. Eles cruzam o projeto ou a implementação de vários ou mesmo muitos blocos de construção e, portanto, são chamados de interesses transversais. Exemplos típicos para questões transversais incluem *log*, sincronização e distribuição. Devido à sua própria natureza, as preocupações transversais implicam dois problemas principais para o desenvolvimento de software. Em primeiro lugar, o seu design ou implementação está espalhado por muitos blocos de construção (o problema de dispersão) e, em segundo lugar, um bloco de construção geralmente compreende o design ou a implementação de mais de uma preocupação (o problema de emaranhamento). (ROSENHAINER, 2004, p. 49, tradução nossa)²

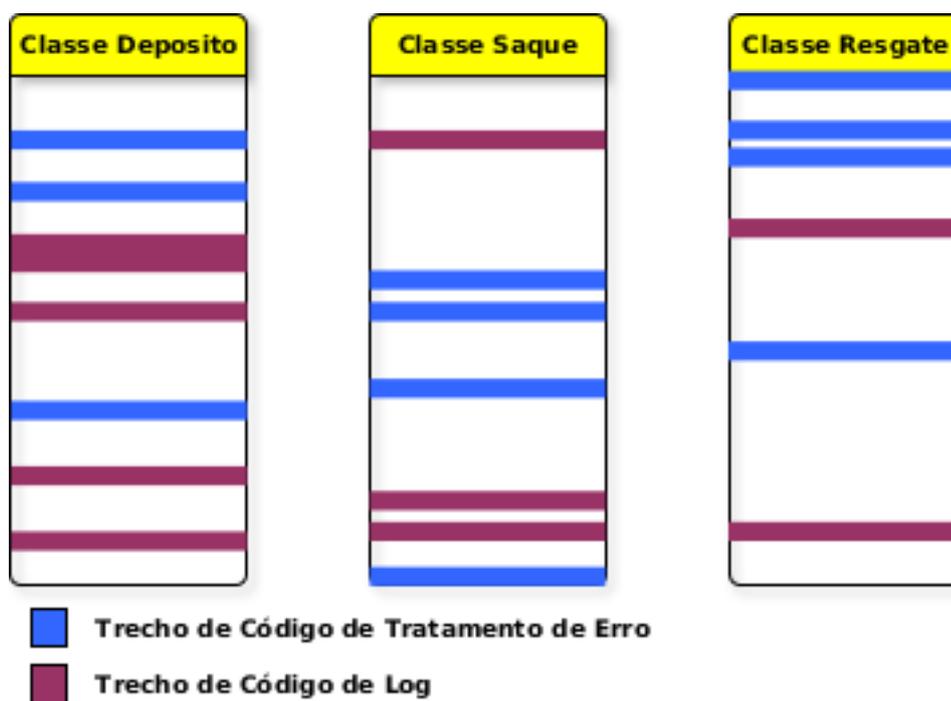
Para ilustrar o comportamento *Crosscutting Concern* no código observa-se na Figura 1 3 (três) classes (Depósito, Saque e Resgate) com ações de códigos para tratamento de erros e códigos de geração de *log*, que por sua vez, não são o objetivo principal das referidas classes.

Na Figura 2 observa-se em uma representação gráfica livre como os códigos para tratamento de erros e os códigos de geração de *log* incidiriam sobre as 3 (três) classes como Aspectos.

¹ Em função do termo *crosscutting concerns* ser muito difundido, na literatura técnica brasileira, usaremos o termo em inglês no decorrer desta dissertação

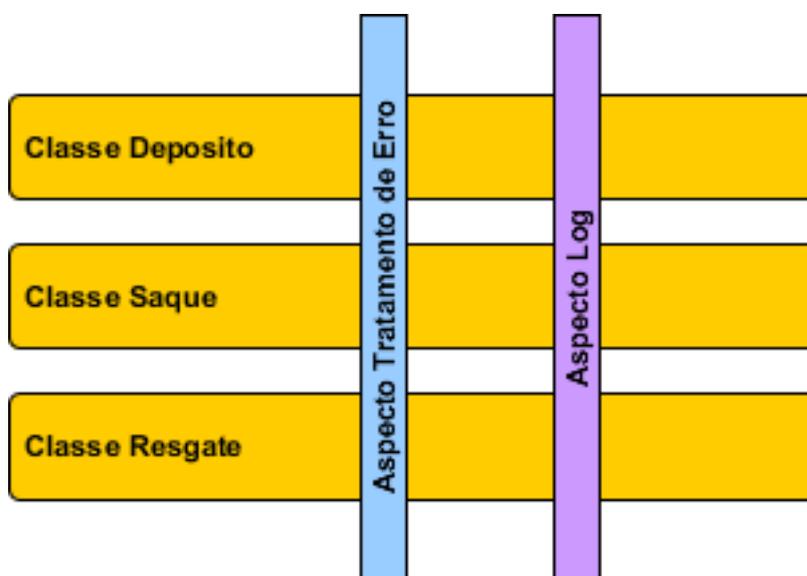
² [Whereas—by means of conventional techniques of modularization or object-orientation—one sort of concerns may smoothly be encapsulated within building blocks such as modules, classes, and operations on design or code level, the same is not possible for another sort of concerns. They crosscut the design or implementation of several or even many building blocks and are therefore called crosscutting concerns. Typical examples for crosscutting concerns include logging, synchronization and distribution. Due to their very nature, crosscutting concerns entail two main problems for software development. First, their design or implementation is scattered over many building blocks (the scattering problem) and second, one building block often comprises the design or implementation of more than one concern (the tangling problem).]

Figura 1 – Exemplo de Classes com *Crosscutting Concerns*



Fonte: Desenvolvido pelo autor.

Figura 2 – Exemplo Aspectos



Fonte: Desenvolvido pelo autor.

Para tratar *crosscutting concern* e outras limitações, ao longo do tempo vem sido criados vários paradigmas que estendem a Orientação a Objetos, tal como a Orientação a Sujeitos, a Orientação a Características, a Orientação a Aspectos, entre outros. Segundo Kiczales *et al.* (1997, p. 2), a Programação Orientada a Aspectos (POA) foi concebida para solucionar o problema do *crosscutting concern* sem abandonar qualquer conceito da

Orientação a Objetos, se conectando ao paradigma principal como uma extensão.

Encontramos muitos problemas de programação para os quais nem as técnicas de programação orientadas a procedimentos nem as orientadas a objetos são suficientes para capturar claramente algumas das importantes decisões de projeto que o programa deve implementar. Isso força a implementação dessas decisões de projeto a serem espalhadas por todo o código, resultando em código "emaranhado" que é excessivamente difícil de desenvolver e manter. Ibid., p. 2, tradução nossa³

Quadro 1 – Frameworks POA

Linguagem de Programação	Framework POA
C# / VB.Net	Aspect.NET Aspect# AspectDNG Encase Spring.NET
C / C++	AspectC AspectC++ FeatureC++
Cobol	Cobble
Cocoa	AspectCocoa
Cold Fusion	ColdSpring
Common Lisp	AspectL
Delphi	InfraAspect
Flash Action Script	as2lib
Java	AspectJ CaesarJ JakartaHivemind Jboss AOP Spring AOP
Java Script	Ajaxpect Aspects AspectJS Jquery AOP Plugin
Lua	AspectLua
PHP	Aspect-Oriented PHP Go! AOP PHP ExarPHP PHP-AOP PHPaspect
Python	AspyctAOP Lightweight Python AOP Spring Python's PHP Module
Ruby	Aquarium AspectR

Fonte: Dados compilados pelo autor.

³ [We have found many programming problems for which neither procedural nor object-oriented programming techniques are sufficient to clearly capture some of the important design decisions the program must implement. This forces the implementation of those design decisions to be scattered through-out the code, resulting in "tangled" code that is excessively difficult to develop and maintain.]

No Quadro 1 estão descritas as principais implementações de POA no mercado distribuídos entre linguagens e *frameworks*.

A POA não tem a sua modelagem contemplada na principal linguagem de modelagem do mercado de desenvolvimento, a Linguagem de Modelagem Unificada (*Unified Modeling Language* - UML), publicada em Rumbaugh, Jacobson e Booch (1999). Em contrapartida muitos estudos tem sido feitos a fim de desenvolver técnicas de modelagem POA (*Aspect Oriented Modeling* (AOM), ou Modelo Orientado à Aspecto)^{4,5}, como:

- *Aspect Oriented - Unified Model Language (AO-UML)*, ou Orientação a Aspecto - Linguagem de Modelagem Unificada;^{6,7}
- *Aspect-Oriented Design Model (AODM)*, ou Modelo de Projeto Orientado à Aspecto;^{8,9}
- **aSideML**;
- *UML All pUrpose Transformer (UMLAUT)*, ou Transformador UML multifuncional;^{10,11}
- *UML eXchange Format, aspect extension (UXF/a)*, ou Formato de intercâmbio em UML com extensão em aspectos;^{12,13}
- **WEAVR**;

O *Design by Contract* (DbC), ou Programação por Contrato¹⁴ é uma técnica criada em Meyer (1992) que descreve a sua aderência a OO no que tange a confiabilidade do sistema a ser desenvolvido, estabelecendo regras, ou "contratos" para uso dos métodos.

Uma das implementações de maior notoriedade no mercado de desenvolvimento de software é a *Java Modeling Language* (JML), ou Linguagem de Modelagem para Java^{15,16},

⁴ Tradução nossa

⁵ Em função do termo *Aspect Oriented Modeling* ser muito difundido na literatura técnica brasileira, usaremos o termo em inglês, bem como sua sigla, no decorrer desta dissertação

⁶ Tradução nossa.

⁷ Em função do termo *Aspect Oriented - Unified Model Language* ser muito difundido na literatura técnica brasileira, usaremos o termo em inglês, bem como sua sigla, no decorrer desta dissertação.

⁸ Tradução nossa.

⁹ Em função do termo *Aspect-Oriented Design Model* ser muito difundido na literatura técnica brasileira, usaremos o termo em inglês, bem como sua sigla, no decorrer desta dissertação.

¹⁰ Tradução nossa.

¹¹ Em função da sigla UMLAUT só fazer algum sentido ao se utilizar o termo em inglês e este ser muito difundido na literatura técnica brasileira, o usaremos no decorrer desta dissertação.

¹² Tradução nossa.

¹³ Em função da sigla UXF/a só fazer algum sentido ao se utilizar o termo em inglês e este ser muito difundido na literatura técnica brasileira, o usaremos no decorrer desta dissertação.

¹⁴ Em função do termo *Design by Contract* ser muito difundido na literatura técnica brasileira, usaremos o termo em inglês, bem como sua sigla, no decorrer desta dissertação.

¹⁵ Tradução nossa.

¹⁶ Em função da sigla JML só fazer algum sentido ao se utilizar o termo em inglês e este ser muito difundido na literatura técnica brasileira, o usaremos no decorrer desta dissertação

que segundo Leavens e Cheon (2006), JML é uma linguagem formal de especificação de interface comportamental para Java que contém as notações essenciais usadas no DbC como um subconjunto próprio.

O paradigma OO sem dúvida representou um avanço na busca pela qualidade de software, porém demorou um pouco para se perceber que apenas sua utilização não era suficiente para construir softwares melhores. Foi necessário então um tempo para que os desenvolvedores assimilassem efetivamente as boas práticas do desenvolvimento orientado a objetos. No entanto, tais práticas - apesar de fundamentais para um bom projeto - não são suficientes por si só para a obtenção de sistemas sem erros. Neste contexto, Design by Contract (DbC) surge como um método que visa a construção de sistemas OO mais confiáveis, na medida em que provê mecanismos para checar a correção de um sistema. (DARIO; GUERRERO, 2005)

Algumas poucas aplicações de DbC que se aplicam aos contratos especificados por Meyer (1992) em elementos do Aspecto foram desenvolvidas ao longo do tempo como:

- **AspectJML** proposta em Rebêlo *et al.* (2013).
- **Barter** proposta em Barter (2002)
- **ConaJ** proposta em Skotiniotis e Lorenz (2004a), Skotiniotis e Lorenz (2004b) e Lorenz e Skotiniotis (2004)
- **Contract4J** proposta em Wampler (2006)
- **ContractAJ** proposta em Molderez e Janssens (2012);
- **Jose** proposto em Feldman, Barzilay e Tyszberowicz (2006);
- **Pipa** proposto em Zhao e Rinard (2003);

1.1 Justificativa da Pesquisa

Uma equipe de desenvolvimento de software pode utilizar a modelagem em UML para previamente definir os detalhes da programação das mais diversas partes do projeto. Se essa equipe adotar algum *framework* de aspectos, como o AspectJ, por exemplo, também é possível definir a estrutura dos aspectos, por intermédio de alguma das técnicas de modelagem publicadas na academia ao longo do tempo, promovendo clareza, objetividade e velocidade, para produção de código em equipe, desenvolvimento em paralelo, tornando as decisões mais ágeis e evitando retrabalhos.

Ainda não é possível modelar, em Diagrama de Classe, a técnica DbC em Classes e Aspectos e é justamente nessa lacuna que a pesquisa se concentra. Com a viabilidade dessa

modelagem, é possível uma documentação, através de uma representação gráfica tanto para orientar a equipe antes do desenvolvimento quanto o registro para consultas futuras, orientar novos integrantes da equipe, auxiliar na refatoração, dentre outros benefícios.

Refatoração é o processo de alteração de um sistema de software de tal forma que não altera o comportamento externo do código ainda melhora sua estrutura interna. É uma maneira disciplinada de limpar código que minimiza as chances de introdução de erros. Em essência, quando você refatorar, você está melhorando o design do código depois de ter sido escrito.(FOWLER *et al.*, 1999, p. 9, tradução nossa)¹⁷

1.2 Motivação

A motivação desse trabalho se dá em função da modelagem de DbC em Classes e Aspectos em Diagrama de Classe da UML ainda ser um problema em aberto.

Os benefícios da criação de uma técnica nesses moldes é muito útil para equipes de desenvolvimento que implementam Aspectos em seu código, independente da linguagem de programação ou do *framework* POA adotado, possibilitando uma maior clareza na prática de desenvolvimento paralelo, com a demonstração das regras das interfaces dos aspectos de forma visual, facilitando a demonstração do projeto para a equipe de desenvolvimento.

Com a criação da técnica de modelar DbC, passa a ser possível o desenvolvimento de módulos que apliquem essa técnica nos sistemas de desenvolvimento UML de mercado e enfim, possibilitar Engenharia Reversa para geração de modelos de sistemas legados (OO / POA / DbC).

1.3 Caracterização do Problema

Os contratos estabelecidos por DbC são eficazes para a delimitação da regra para uso do método, porém não há como modelá-los no padrão do Diagrama de Classe estabelecido pela UML, devido ao fato da UML ser restrita a OO e DbC ser uma extensão deste paradigma.

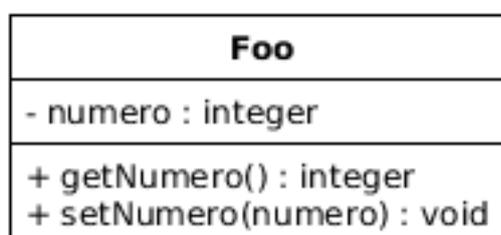
Por exemplo, na Figura 3 observa-se a classe Foo com um atributo numero e 2 (dois) métodos (getNumero e setNumero). Caso houvesse um contrato DbC para o método setNumero, em que só fosse permitido valores menores que 10 (dez), conforme código java abaixo, a representação do contrato no modelo da classe não seria possível.

```
1 public class Foo {  
2     private integer numero;
```

¹⁷ [Refactoring is the process of changing a software system in such a way that it does not alter the external behavior of the code yet improves its internal structure. It is a disciplined way to clean up code that minimizes the chances of introducing bugs. In essence when you refactor you are improving the design of the code after it has been written.]

```
3
4     public integer getNumero() {
5         return this.numero;
6     }
7
8     //@ requires valor < 10;
9     public void setNumero(integer numero) {
10        this.numero = numero;
11    }
12 }
```

Figura 3 – Exemplo Classe



Fonte: Desenvolvido pelo autor.

O problema de não se ter como modelar DbC em Diagramas de Classes se apresenta ainda maior quando se aplica DbC em Aspectos, como em AspectJML, pois em função da POA não estar contida na UML, alguns estudos foram feitos nesse sentido e algumas extensões para a UML foram elaboradas ao longo do tempo. Ainda hoje não há um consenso sobre uma técnica que tivesse virado um padrão de mercado, como a UML para a OO, por exemplo.

A falta de uma técnica de modelagem de DbC prejudica equipes de desenvolvimento de software que utilizam esta tecnologia devido ao fato de não ser possível uma representação gráfica do sistema que está sendo produzido, seja para orientar o desenvolvimento, seja para documentar a solução, restando apenas o próprio código como documentação do comportamento das regras que contemplam o Aspecto.

1.3.1 Abordagem do Problema

Em relação a abordagem do problema, utiliza-se a pesquisa qualitativa de uma técnica se seja possível modelar Classes e Aspectos considerando a técnica de *Design by Contract* em ambos.

1.4 Natureza da Pesquisa

Utilizamos neste trabalho a pesquisa aplicada, pois o trabalho objetiva gerar um padrão de modelo de Diagrama de Classe com *Design by Contract* sobre Classes e Aspectos que poderá ser implementado em qualquer sistema de mercado que gere modelos UML. O padrão de modelo supracitado é fruto da avaliação da técnica DbC e nos estudos de modelagem existentes e publicados em artigos científicos, teses e dissertações. Para desenvolver a técnica de modelagem, optou-se pela sintaxe de DbC sobre Aspectos do AspectJML e o padrão de modelagem de Aspectos em AODM.

1.5 Questão da Pesquisa

Com base na descrição do problema, segue a questão central da pesquisa:

Como definir uma técnica de modelagem de Diagrama de Classe, com base na UML e em uma extensão para POA, que possibilite a representação gráfica da técnica *Design by Contract*?

1.6 Objetivos

Nesta seção, encontram-se a apresentação do objetivo geral e dos objetivos específicos da dissertação. A pesquisa é descritiva, com o propósito de discorrer sobre as técnicas estudadas, bem como o modelo de *Design by Contract* a ser definido, possibilitando sua implementação real em Classes e Aspectos.

1.6.1 Objetivo Geral

O objetivo desta dissertação é complementar a notação de modelo de Diagrama de Classe/UML com Aspectos, de modo a abranger a técnica de *Design by Contract*.

1.6.2 Objetivo Específico

A fim de resolver o problema exposto nesta dissertação e alcançar o objetivo geral, temos como objetivo específico:

- Desenvolver uma técnica para modelar DbC sobre Classes e utilizando uma extensão de Aspectos para Diagrama de Classe da UML, possibilitando a demonstração das características dos aspectos de forma visual e facilitar a demonstração do projeto para a equipe de desenvolvimento;

1.7 Organização da Dissertação

O Capítulo 1 - Introdução - tem um caráter introdutório e ambientador em relação ao tema. Este capítulo também aborda a caracterização do problema, bem como uma descrição da contribuição do trabalho para o desenvolvimento de software.

O Capítulo 2 - Metodologia - discorre sobre o paradigma e a metodologia escolhidos para o desenvolvimento desta dissertação, as justificativas das escolhas, bem como uma descrição dos critérios para a condução da pesquisa e as suas etapas.

O Capítulo 3 - Revisão Sistemática da Literatura - (RSL), apresenta a estratégia de revisão, critérios e termos de busca e o resultado da consulta às bases de conhecimento a fim de delinear o estado da arte dos assuntos inerentes a esta pesquisa.

O Capítulo 4 - Abordagem das Técnicas Base - descreve uma apresentação das tecnologias, com base no levantamento feito pela Revisão Sistemática da Literatura, que serve de base para o desenvolvimento do artefato que expresse contratos estabelecidos pela DbC em classes e aspectos. As tecnologias expostas são AODM, JML e AspectJML.

O Capítulo 5 - Projeto do Artefato - trata da abordagem da modelagem do artefato desenvolvido pela pesquisa, bem como um exemplo de aplicação prática e a avaliação do artefato.

O Capítulo 6, e último - Conclusão - apresenta de maneira sintética um apanhado geral do desenvolvimento da pesquisa e do artefato com uma conclusão do trabalho, bem como as propostas de trabalhos futuros.

2 METODOLOGIA

A apresentação da metodologia desta dissertação inicia-se com a descrição do paradigma que envolve a proposta de solução do problema em questão. Após a descrição do paradigma, segue-se uma explicação acerca do método de pesquisa escolhido, juntamente com a descrição de seus componentes e etapas. O capítulo finaliza com uma inter-relação entre as etapas do método de pesquisa e a distribuição dos seus capítulos.

2.1 Paradigma *Design Science*

O foco desta dissertação é o desenvolvimento de uma técnica no qual seja possível a representação gráfica de DbC sobre Classes e Aspectos em Diagramas de Classes, respeitando padrões existentes em um dos diversos trabalhos publicados de modelagem de Aspectos com base na estrutura da UML. Sendo assim, como resultado do estudo, é apresentada uma técnica de modelagem, culminando na geração de um artefato do tipo modelo que estende um outro já existente.

Um artefato é algo criado por pessoas para algum propósito prático. Exemplos de artefatos projetados e estudados em sistemas de informação e pesquisa de engenharia de software são algoritmos, métodos, notações, técnicas e até mesmo estruturas conceituais. Eles são usados na concepção, desenvolvimento, implementação, manutenção e utilização de sistemas de informação e sistemas de software. Quando um artefato é usado, ele é usado por pessoas, o que significa que ele interage com um contexto que, junto com outras coisas, contém pessoas. (WIERINGA, 2014, p. 29, tradução nossa)¹

Em função do exposto, foi feita uma busca para a escolha de um paradigma e metodologia para a construção da dissertação. Como resultado dessa busca foram observados a metodologia CommonKADS, proposta por Schreiber *et al.* (2000), CEM (Composition, Environment, Structure, Mechanism, ou Composição, Ambiente, Estrutura, Mecanismo)^{2,3}, proposta por Bunge (2003) e a *Design Science* (DS), ou Ciência do Artificial⁴, que foi um conceito introduzido no meio científico através de Simon (1996), cuja primeira edição é

¹ [An artifact is something created by people for some practical purpose. Examples of artifacts designed and studied in information systems and software engineering research are algorithms, methods, notations, techniques, and even conceptual frameworks. They are used when designing, developing, implementing, maintaining, and using information systems and software systems. When an artifact is used, it is used by people, which means that it interacts with a context that, along with other things, contains people.]

² Tradução nossa.

³ Em função do termo *Composition, Environment, Structure, Mechanism* ser muito difundido na literatura técnica brasileira, usaremos o termo em inglês, bem como sua sigla, no decorrer desta dissertação.

⁴ Em função do termo *design science* ser muito difundido, na literatura técnica brasileira, usaremos o termo em inglês no decorrer desta dissertação

datada de 1969, e posteriormente corroborado por Walls, Widmeyer e Sawy (1992), Gibbons *et al.* (1994), March e Smith (1995), Romme (2003), Van Aken (2004), Van Aken (2005).

O paradigma escolhido para a orientação desta dissertação foi a *Design Science*, que é classificada como um paradigma adequado para trabalhos que visam a criação de novos artefatos, ou a elaboração de um artefato que altera algum outro já existente. "O paradigma *design science* procura ampliar os limites das capacidades humanas e organizacionais criando artefatos novos e inovadores". (HEVNER *et al.*, 2004, p. 1, tradução nossa)⁵

Acima de tudo, a *design science* é a ciência que procura desenvolver e projetar soluções para melhorar sistemas existentes, resolver problemas ou ainda, criar novos artefatos que contribuam para uma melhor atuação humana, seja na sociedade, seja nas organizações. (DRESCH; LACERDA; ANTUNES JÚNIOR, 2015, p. 57)

A *design science*, por sua vez, seria responsável por conceber e validar sistemas que ainda não existem, seja criando, recombinao ou alterando produtos / processos / software / métodos para melhorar as situações existentes. *Ibid.*, p. 60

A *design science* é, sem dúvida, uma abordagem que pode orientar pesquisas que se destinam a projetar ou desenvolver algo novo, uma vez que a *design science* tem como foco causar a mudança, criando artefatos e gerando soluções para problemas existentes. *Ibid.*, p. 123

Uma característica importante da *Design Science*, em relação a resolução do problema, é que este paradigma não visa encontrar uma solução que seja ótima, ideal ou definitiva, e sim uma "solução satisfatória" e funcional em relação ao contexto avaliado. "É claro que a decisão que é ideal para a aproximação simplificada raramente será o ideal no mundo real, mas a experiência mostra que ela será muitas vezes satisfatória." (SIMON, 1996, p. 27, tradução nossa)⁶

A "solução satisfatória" é válida desde que seja permitida a generalização para uma classe de problemas, gerando assim, conhecimento de modo a ser possível a execução de testes, simulações ou reuso da técnica produzida, por outros pesquisadores e profissionais, possibilitando sua reprodução em cenários e situações diversas. "Um certo grau de generalização é reconhecido como um pré-requisito para a teoria, mesmo amplamente definida"(GREGOR; JONES, 2007, p. 15, tradução nossa)⁷

A classe de problemas é uma abstração de um problema para uma forma mais generalista, de modo a ser possível a identificação de padrões, a criação de regras para se solucionar problemas classificados de um mesmo tipo. "(...) as classes de problemas são

⁵ [The design-science paradigm seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts.]

⁶ [Of course the decision that is optimal for the simplified approximation will rarely be optimal in the real world, but experience shows that it will often be satisfactory.]

⁷ [The degree to which design knowledge can be expressed in general propositions remains an issue. Some degree of generality is recognized as a prerequisite for theory, even broadly defined.]

arquétipos necessários quando um novo modelo de produção de conhecimento é abordado. A discussão principal envolve essa necessidade de definir as classes de problemas."(VEIT *et al.*, 2017 apud VAN AKEN, 2004, p. 298, tradução nossa)⁸ "As classes de problemas não permitem que os artefatos e, conseqüentemente, suas soluções respondam oportunamente a algum problema em um contexto particular."(VEIT *et al.*, 2017 apud DRESCH; LACERDA; ANTUNES JÚNIOR, 2015, p. 298, tradução nossa)⁹

Outra característica fundamental da *Design Science Research* é que, embora orientada para a resolução de problemas, não busca uma solução ótima, mas sim uma solução satisfatória para os problemas em estudo. Além disso, embora o problema abordado seja único e específico, as soluções obtidas a partir da condução de *Design Science Research* devem ser capazes de generalização para uma determinada classe de problemas. (DRESCH; LACERDA; MIGUEL, 2015, p. 9, tradução nossa)¹⁰

2.2 Método de Pesquisa *Design Science Research*

O método de pesquisa escolhido para esta dissertação foi o *Design Science Research Methodology* ("Metodologia de Pesquisa da Ciência do Artificial")¹¹, ou simplesmente *Design Science Research* (DSR), ou ainda, "Pesquisa da Ciência do Artificial"¹² (como os termos *Design Science Research Methodology* e *Design Science Research*, não possuem tradução consensual na literatura técnica brasileira, usaremos os termos em inglês no decorrer desta dissertação).

A metodologia *Design Science Research* surgiu de conceitos que vinham evoluindo desde os anos 1980 e teve o seu nome consolidado através de publicações como as de March e Smith (1995), Hevner *et al.* (2004), Vaishnavi e Kuechler (2004) e Peffers *et al.* (2007), em que foi apresentada a sua estrutura metodológica aderente ao paradigma da *Design Science*, incorporando princípios, práticas e procedimentos necessários para prover subsídios para atender a sua principal característica, que vem a ser a criação de novos artefatos ou a evolução de um artefato já existente.

A *design science* é a base epistemológica¹³ quando se trata do estudo do que é artificial. A *design science research*, por sua vez, é o método que

⁸ [(...) the classes of problems are a necessary agenda when a new knowledge production model is approached. The main discussion involves this need to define the classes of problems.]

⁹ [The classes of problems do not allow the artifacts and, consequently, their solutions to respond timely to some problem in a particular context]

¹⁰ [Another key characteristic of Design Science Research is that, although it is oriented to problem solving, it does not seek an optimal solution but rather a satisfactory solution to the problems under study. Furthermore, although the addressed problem is unique and specific, solutions obtained from the conduction of Design Science Research must be capable of generalization to a certain class of problems]

¹¹ Tradução do autor

¹² Tradução do autor

¹³ "'Epistemologia' significa discurso (logos) sobre a ciência (episteme). (Episteme + logos). Epistemologia: é a ciência da ciência. Filosofia da ciência. É o estudo crítico dos princípios, das hipóteses e dos resultados das diversas ciências. É a Teoria do Conhecimento."(TESSER, 1994)

fundamenta e operacionaliza a condução da pesquisa quando o objetivo a ser alcançado é um artefato ou uma prescrição. Como método de pesquisa orientado à solução de problemas, a *design science research* busca, a partir do entendimento do problema, construir e avaliar artefatos que permitam transformar situações, alterando suas condições para estados melhores ou desejáveis. Ela é utilizada nas pesquisas como forma de diminuir o distanciamento entre teoria e prática. (DRESCH; LACERDA; ANTUNES JÚNIOR, 2015, p. 67)

A partir dessa análise comparativa, podemos constatar que a *design science research* é o método de pesquisa mais indicado quando o objetivo do estudo é projetar e desenvolver artefatos, bem como soluções prescritivas, seja em um ambiente real ou não. *Ibid.*, p. 95

Vaishnavi e Kuechler (2004) afirmam que a metodologia *Design Science Research* pode produzir 8 (oito) tipos de artefatos através de trabalhos de pesquisa científica, gerando assim bases de conhecimento que são: Constructos, Modelos, *Frameworks*, Arquiteturas, Princípios de Projeto, Métodos, Instâncias e Teorias de Projeto, conforme descrito no Quadro 2. Alguns autores como March e Smith (1995) e Peffers *et al.* (2007) agrupam os artefatos supracitados, de maneira mais compacta, em apenas 4 (quatro) tipos de artefatos (construtos, modelos, métodos e instâncias), como se pode observar na mesma tabela.

Quadro 2 – Tipos de Artefatos da *Design Science Research*

ARTEFATO	DESCRIÇÃO
Constructos	Vocabulário conceitual de um domínio. "Constructos ou conceitos formam o vocabulário de um domínio. Eles constituem uma conceituação usada para descrever problemas dentro do domínio e para especificar suas soluções. Eles formam a linguagem especial e o conhecimento compartilhado de uma disciplina ou subdisciplina."(MARCH; SMITH, 1995, p. 100, tradução nossa) ¹ Ou ainda, "Eles definem os termos usados ao descrever e pensar sobre tarefas."Ibid., p. 256, tradução nossa ²
Modelos	Conjuntos ou proposições ou declarações expressando relações entre constructos.
<i>Frameworks</i>	Guias reais ou conceituais para servir de suporte.
Arquiteturas	Estruturas de sistemas de alto nível.
Princípios de Projeto	Princípios e conceitos básicos para orientar o projeto.
Métodos	Conjuntos de etapas usadas para executar tarefas como conhecimento.
Instancias	Implementações situadas em ambientes que fazem ou não operações em constructos, modelos, métodos e outros artefatos abstratos; Neste último caso, esse conhecimento permanece tácito.
Teorias de Projeto	Um conjunto prescritivo de declarações sobre como fazer algo para alcançar um determinado objetivo. Uma teoria geralmente inclui outros artefatos abstratos, tais como constructos, modelos, <i>frameworks</i> , arquiteturas, princípios de projetos e métodos.

¹ [Constructs or concepts form the vocabulary of a domain. They constitute a conceptualization used to describe problems within the domain and to specify their solutions. They form the special ized language and shared knowledge of a discipline or subdiscipline.]

² [They define the terms used when describing and thinking about tasks.]

Segundo Hevner *et al.* (2004), para a condução de pesquisas que utilizam o método *Design Science Research*, deve-se cumprir rigorosamente 7 (sete) critérios, a fim de garantir sua validade e confiabilidade, conforme descrito na Quadro 3. Sordi, Meireles e Sanches (2010) afirmam que essas 7 (sete) diretrizes se tornaram referências para pesquisadores, revisores, editores e para o público geral que necessita compreender e avaliar um método de pesquisa científica sob o paradigma da *Design Science*, em especial para a área de Sistemas de Informação. "Tais diretrizes devem ser criteriosamente observadas em qualquer pesquisa que adote a abordagem *design science*." *Ibid.*, p. 4

Quadro 3 – Critérios para a condução das pesquisas que utilizam a *Design Science Research*

#	Critério	Descrição
1	<i>Design</i> como artefato	As pesquisas desenvolvidas com o método da <i>Design Science Research</i> devem produzir artefatos viáveis, na forma de um constructo, modelo, método ou instanciação.
2	Relevância do problema	O objetivo da <i>Design Science Research</i> é desenvolver soluções para resolver problemas importantes e relevantes para as organizações.
3	Avaliação do <i>Design</i>	A utilidade, a qualidade e a eficácia do artefato devem ser, rigorosamente demonstradas por meio de métodos de avaliação bem executados.
4	Contribuições da pesquisa	Uma pesquisa conduzida pelo método da <i>Design Science Research</i> deve prover contribuições claras e verificáveis nas áreas específicas dos artefatos desenvolvidos e apresentar fundamentação clara em fundamentos de <i>design</i> e/ou metodologias de <i>design</i> .
5	Rigor da pesquisa	A pesquisa deve ser baseada na aplicação de métodos rigorosos, tanto na construção como na avaliação dos artefatos.
6	<i>Design</i> como um processo de pesquisa	A busca por um artefato efetivo exige a utilização de meios que estejam disponíveis para alcançar os fins desejados, ao mesmo tempo que satisfaçam as leis que regem o ambiente em que o problema está sendo estudado.
7	Comunicação da pesquisa	As pesquisas conduzidas pela <i>Design Science Research</i> devem ser apresentadas tanto para o público mais orientado à tecnologia quanto para aquele mais orientado à gestão.

Fonte: (DRESCH; LACERDA; ANTUNES JÚNIOR, 2015 apud HEVNER *et al.*, 2004, p. 70)

2.3 Etapas para a condução da pesquisa

A metodologia DSR se faz adequada para esta dissertação principalmente por moldar com propriedade duas características proeminentes deste trabalho: a condução de abordagem qualitativa e a orientação da produção de artefatos.

Uma das características marcantes da metodologia é que ela se utiliza em uma de suas etapas mais importantes o Método Abdução, que traz um comportamento moderno

para a metodologia ao se permitir a criar novos artefatos. E ainda, para consolidar a sua condução, em outras etapas, se utiliza do Método Dedutivo para desenvolver e avaliar o artefato e do Método Indutivo para estabelecer a generalização para uma classe de problemas. O Método Hipotético-Dedutivo, não é utilizado na *Design Science Research*, por ela não possuir um caráter de refutação de ideias anteriores, pelo contrário, a criação é uma das características mais marcantes da metodologia.

Como uma forma de conduzir a dissertação conforme o modelo da *Design Science Research* foi adotado o método de condução proposto por Dresch, Lacerda e Antunes Júnior (2015)¹⁴, que tem uma estrutura clara, objetiva, composta de 12 (doze) etapas principais. O fluxo entre as etapas é bem estabelecido, conforme Figura 4, em que as setas contínuas indicam o fluxo da realização das etapas e as setas tracejadas os possíveis retornos entre as etapas.

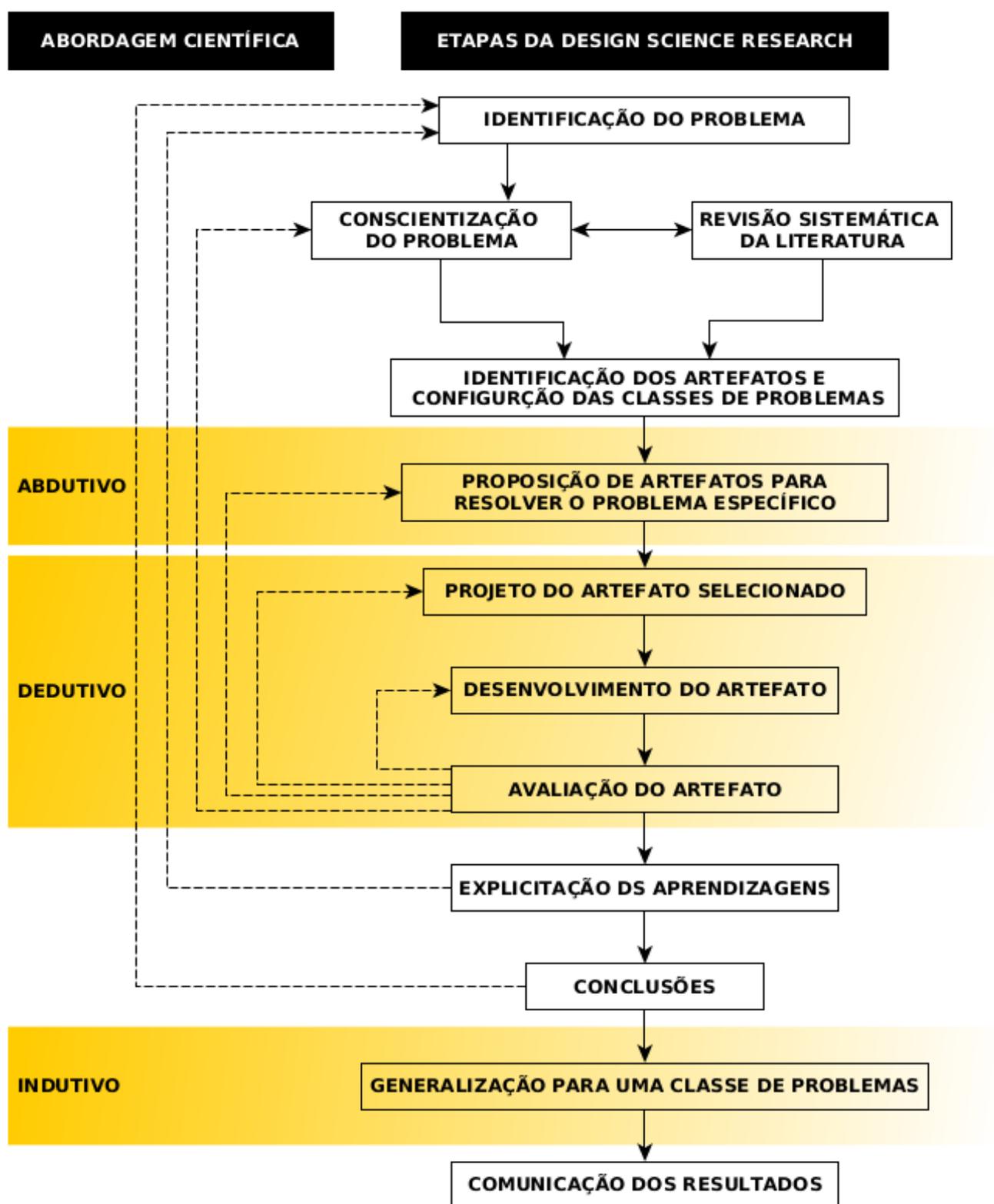
2.3.1 Etapa da *Design Science Research* x Estrutura da Dissertação

A aderência entre as etapas da *Design Science Research*, apresentadas na Seção 2.3 e a produção deste trabalho, que está distribuída em capítulos, cuja organização já foi apresentada na Seção 1.7 não está construída de maneira linear e uniforme a partir deste ponto, nem utiliza os termos das etapas aqui apresentadas. Portanto, para orientar a aplicação do método *Design Science Research* nesta dissertação, e assim relacionar todos os seus elementos e o que deve-se ter de produto (saída) em cada uma das etapas, foi elaborado o Quadro 4.

Ainda com o intuito de orientar a leitura e compreensão do modo de elaboração deste trabalho, e assim, fundamentar sua concepção, com base na orientação do método *Design Science Research*, foi formalizado um protocolo de pesquisa que está disponível no Apêndice A. Neste protocolo estão apresentadas as atividades realizadas durante a pesquisa em cada 1 (uma) das 12 (doze) etapas da *Design Science Research*.

¹⁴ Toda a seção "Etapas para a condução da pesquisa", bem como suas seções foram elaboradas baseadas no estudo de Dresch, Lacerda e Antunes Júnior (2015)

Figura 4 – Método para condução da *Design Science Research*



Fonte: Desenvolvido pelo autor, baseado em (DRESCH; LACERDA; ANTUNES JÚNIOR, 2015, p. 125)

Quadro 4 – Etapas da *Design Science Research* x Estrutura de Capítulos da Dissertação

Nº Etapa	Etapa da Metodologia	Capítulos	Produto
1ª Etapa	Identificação do problema	Seção 1.5	Questão de pesquisa formulada
2ª Etapa	Conscientização do Problema	Capítulos 1 e 3	Descrição detalhada do problema e dos limites das tecnologias existentes.
3ª Etapa	Revisão sistemática da literatura	Capítulo 3	Formalização das faces do problema, compreensão do ambiente externo, requisitos do artefato e revisão sistemática da literatura.
4ª Etapa	Identificação dos artefatos e configuração das classes de problemas	Seção 1.4	Artefatos identificados, classes de problemas estruturadas e configuradas, e soluções satisfatórias explicitadas
5ª Etapa	Proposição de artefatos para resolver o problema específico	Secao 5.1	Formalização do artefato para a pesquisa
6ª Etapa	Projeto do artefato selecionado	Seções 5.1, 5.3, Anexos A e B	Projeto explicitando técnicas e ferramentas para o desenvolvimento e a avaliação do artefato, e detalhamento dos requisitos do artefato
7ª Etapa	Desenvolvimento do artefato	Seção 5.2	Heurísticas de construção: artefato em seu estado funcional
8ª Etapa	Avaliação do artefato	Seção 5.3 e Seção 5.3.1	Heurísticas contingenciais: artefato avaliado
9ª Etapa	Explicitação das aprendizagens	Seção 5.3 e Anexos A e B	Aprendizagens formalizadas
10ª Etapa	Conclusões	Capítulo 6	Resultados da pesquisa, principais decisões tomadas e limitações encontradas
11ª Etapa	Generalização para uma classe de problemas	Capítulo 6	Generalização das heurísticas de construção e contingencias para uma classe de problemas
12ª Etapa	Comunicação dos resultados	Capítulo 6	Publicação em <i>journals</i> , revistas setoriais, seminários, congressos, etc.

Fonte: Desenvolvido pelo autor com base em (DRESCH; LACERDA; ANTUNES JÚNIOR, 2015)

3 REVISÃO SISTEMÁTICA DA LITERATURA

A metodologia *Design Science Research* adota a Revisão Sistemática da Literatura na sua terceira etapa para a construção da pesquisa como forma de busca do conhecimento para se estabelecer o estado da arte do problema estudado, como já visto na Seção 2.3.

Segundo Van Aken (2005) para a Revisão Sistemática da Literatura, deve-se buscar publicações especializadas na área de estudo; logo, para tal finalidade nesta dissertação, é adotado o modelo desenvolvido por Kitchenham e Charters (2007), que elenca algumas razões mais comuns para sua realização:

- Resumir as evidências existentes sobre um tratamento ou tecnologia em busca dos benefícios e limitações do objeto em estudo;
- Identificar lacunas do que está sendo estudado, a fim de sugerir áreas para uma investigação mais aprofundada;
- Fornecer um modelo estrutural para direcionar adequadamente novas atividades de pesquisa;
- Examinar até que ponto a evidência empírica suporta ou mesmo contradiz hipóteses teóricas;
- Auxiliar na geração de novas hipóteses.

Uma revisão sistemática da literatura (muitas vezes referida como uma revisão sistemática) é um meio de identificar, avaliar e interpretar todas as pesquisas disponíveis relevantes para uma determinada questão de pesquisa, área de tópico ou fenômeno de interesse. Estudos individuais que contribuem para uma revisão sistemática são chamados estudos primários; Uma revisão sistemática é uma forma de estudo secundário. Ibid., p. 3, tradução nossa¹

O referido modelo de Revisão Sistemática da Literatura propõe um primeiro nível de divisão de sua estrutura em 3 (três) etapas, que por sua vez se subdividem em atividades como pode ser visto na Quadro 5 e na Figura 5.

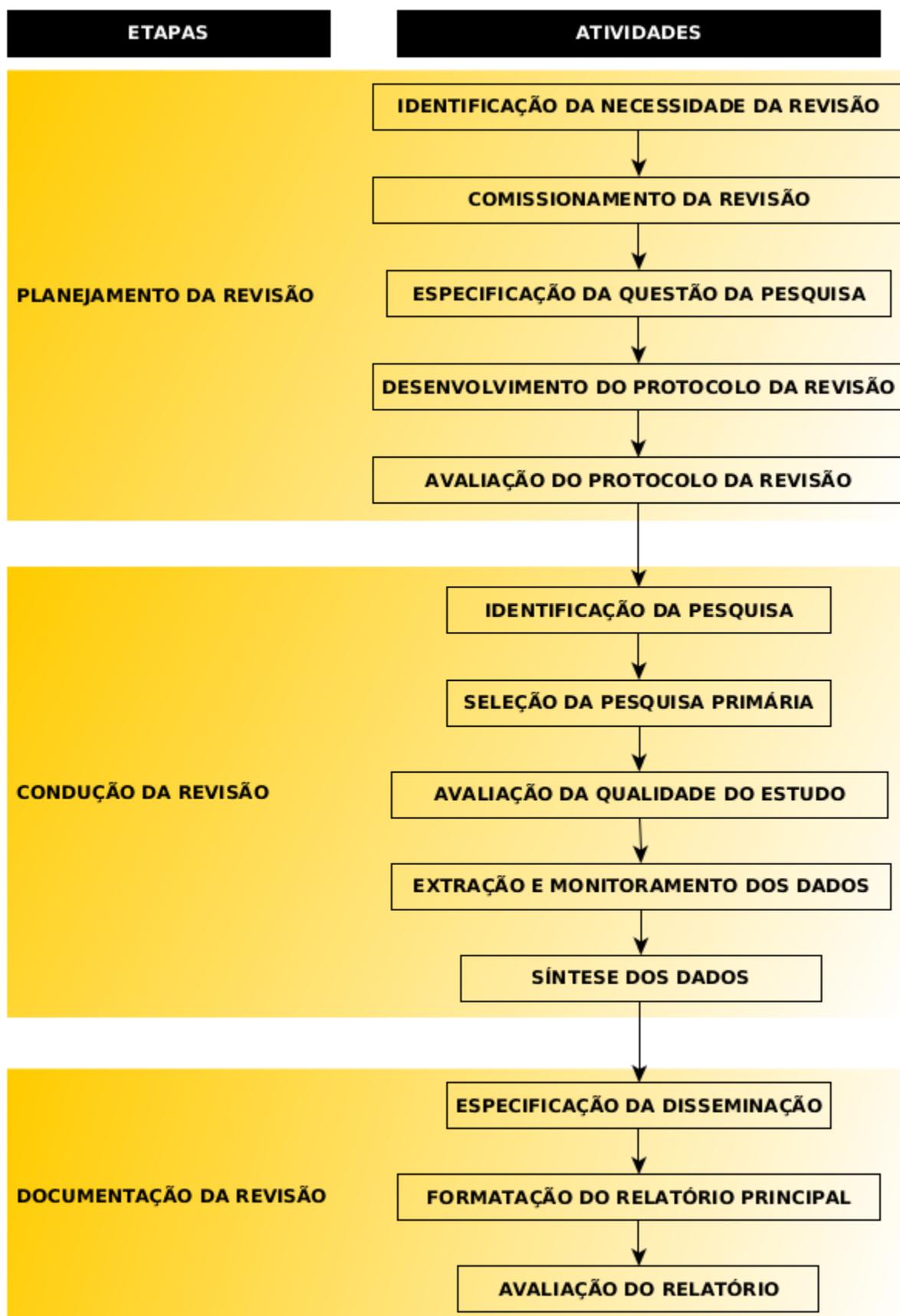
¹ [A systematic literature review (often referred to as a systematic review) is a means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest. Individual studies contributing to a systematic review are called primary studies; a systematic review is a form of secondary study.]

Quadro 5 – Etapas e Atividades da Revisão Sistemática da Literatura

Etapa	Atividade	Descrição
Planejamento da revisão	Identificação da necessidade da revisão	Descrever a necessidade percebida pelo estudo prévio que justifique a Revisão Sistemática da Literatura.
	Comissionamento da revisão	Comissionamento feito a terceiros para desempenhar o papel de pesquisadores e realizarem a Revisão Sistemática da Literatura.
	Especificação da questão da pesquisa	A questão da pesquisa dirige toda a metodologia de Revisão Sistemática (podendo ser a mesma descrita na introdução da pesquisa).
	Desenvolvimento do protocolo da revisão	Montar a estrutura da Revisão Sistemática da Literatura com todas as instruções para prover a revisão, de modo a possibilitar uma reprodução por outros pesquisadores.
	Avaliação do protocolo de revisão	Deverá haver um especialista, que não o desenvolvedor do protocolo, para avaliar a sua pertinência.
Condução da revisão	Identificação da pesquisa	Determinar e seguir uma estratégia de busca estabelecendo critérios como "horizonte temporal", "idiomas", "tipo de revisão", "termos de busca" e "fontes de busca".
	Seleção da pesquisa primária	Seleção dos estudos primários potencialmente relevantes que foram obtidos na identificação da pesquisa com base em critérios de inclusão e exclusão.
	Avaliação da qualidade do estudo	Estabelecer critérios mais refinados para avaliar os estudos primários, além dos critérios de seleção.
	Extração e monitoramento dos dados	Projetar formulários de extração de dados para registrar com precisão a informação obtida nos estudos primários.
	Síntese dos dados	Montar um resumo dos resultados dos estudos primários de forma qualitativa ou quantitativa
Documentação da revisão	Especificação da disseminação	Elaborar uma estratégia de disseminação dos resultados da revisão.
	Formatação do relatório principal	Estabelecer o formato do relatório que serão divulgados os resultados da revisão.
	Avaliação do relatório	Submeter os resultados da revisão para uma banca de pesquisadores, com o propósito de obter uma avaliação da revisão.

Fonte: Baseado em Kitchenham e Charters (2007)

Figura 5 – Método para condução da Revisão Sistemática da Literatura



Fonte: Desenvolvido pelo autor, baseado em Kitchenham e Charters (2007)

3.1 Planejamento da Revisão

A UML, mantida pela OMG (*Object Management Group*, ou "Grupo de Gerenciamento de Objetos")^{2,3}, oferece uma gama de diagramas que modelam a Orientação a Objetos; porém, com estes diagramas não é possível modelar as extensões da OO. Desde o surgimento da POA, alguns trabalhos foram publicados com técnicas que tornam possível a sua modelagem; porém, nenhuma dessas propostas abrangem a técnica DbC.

Sendo assim, equipes de desenvolvimento de software que utilizam a modelagem em UML para previamente definir os detalhes da programação, mesmo adotando um dos padrões de modelagem POA publicados, não conseguem representar graficamente os contratos, nem em Classes nem em Aspectos, perdendo clareza, objetividade e velocidade para a produção de código em equipe, prejudicando o desenvolvimento em paralelo e dificultando a tomada de decisões, correndo o risco de provocar retrabalhos, dentre outros prejuízos.

Um modelo de um sistema de software ajuda os desenvolvedores a explorar várias arquiteturas e soluções de design facilmente antes de escrever o código. Uma boa linguagem de modelagem permite que o designer obtenha a arquitetura geral antes do início do projeto detalhado. (RUMBAUGH; JACOBSON; BOOCH, 1999, p. 14, tradução nossa)⁴

Com base nessa lacuna percebida, a pesquisa se concentra em responder a seguinte questão: Como definir uma técnica de modelagem de Diagrama de Classe, com base na UML e em uma extensão para POA, que possibilite a representação gráfica da técnica *Design by Contract*?

A descrição de todos os detalhes de como foi desenvolvida a revisão para se alcançar este resultado, com base em Kitchenham e Charters (2007), está descrita no Protocolo da Revisão Sistemática da Literatura que se encontra no Apêndice B.

(...), o protocolo fornece detalhes do plano de revisão, incluindo, por exemplo, a especificação do processo a ser seguido e as condições a serem aplicadas na seleção dos estudos primários, as métricas de qualidade a serem aplicadas aos estudos primários e a alocação de revisores a atividades específicas. (BRERETON *et al.*, 2007, p. 576, tradução nossa)⁵

² Tradução nossa.

³ Em função do termo *Object Management Group* ser muito difundido na literatura técnica brasileira, usaremos o termo em inglês, bem como sua sigla, no decorrer desta dissertação.

⁴ [A model of a software system helps developers explore several architectures and design solutions easily before writing code. A good modeling language allows the designer to get the overall architecture right before detailed design begins.]

⁵ [(...), the protocol gives details of the plan for the review, including, for example, specifying the process to be followed and the conditions to apply when selecting primary studies, the quality metrics to be applied to primary studies and the allocation of reviewers to particular activities.]

3.2 Condução da Revisão

Em função de estudos importantes em relação a modelagem da POA terem sido publicados no final dos anos 1990, decidiu-se por uma amplitude do horizonte temporal de 20 (vinte) anos. Sendo assim, as buscas se concentraram em trabalhos publicados a partir de 1997 nos idiomas português, devido ser o idioma nativo, ou em inglês, em função da grande quantidade de bases de dados, periódicos e conferências científicas que publicam neste idioma.

Como a questão elaborada para motivar a pesquisa objetiva responder e resolver um problema que encontra-se em aberto, adotou-se a "Revisão Configurativa" que, segundo Sandelowski *et al.* (2012), tende a ser respondida com dados qualitativos, extraídos de estudos primários heterogêneos.

A fim de concentrar a pesquisa em fontes de buscas relevantes no meio científico e orientar outros pesquisadores que porventura podem vir a se utilizar desta pesquisa, foram selecionadas as seguintes fontes:

Biblioteca Digital Brasileira de Teses e Dissertações: <http://bdtd.ibict.br/vufind/>

EBSCOhost: <https://www.ebscohost.com>

Google Scholar: <https://scholar.google.com.br>

SciELO: <http://www.scielo.org>

Scopus: <http://www.scopus.com>

O termo padrão de busca foi elaborado levando em consideração os idiomas português e inglês simultaneamente; porém, como ele está baseado em siglas, que são comuns a outras áreas, como saúde, geografia, educação, entre outras, optou-se por um bloco de restrição (NOT) dentro do termo de busca, a fim de melhorar a qualidade do resultado. Em contrapartida, foi incluído explicitamente o termo "Motorola" porque a referida empresa possui uma implementação de modelagem de aspectos.

Como as implementações nas fontes de dados operam com sistemas diferentes, e há diferenças na sintaxe, principalmente em relação aos operadores de proximidade, como o "NEAR" utilizado no termo padrão de busca, foi necessário uma adaptação do termo para cada fonte de dados, como a seguir:

TERMO PADRÃO DE BUSCA: (((model* OR UML OR Motorola) NEAR/3 (Aspect* OR AOP OR POA)) OR "Design by Contract") NOT (mathemat* OR hospit* OR biolog* OR health* OR medic* OR *chemic* OR marketing OR food OR restaurant OR drink OR Meteorolog* OR Psycholog* OR electr* OR merchandising OR social OR temperature

OR physics OR mechatronic* OR philosoph* OR alcohol OR mechanic* OR physiolog* OR Intertextuality OR acoustic OR pollution OR pedagog* OR "verb phrase"OR gis OR telecommunication OR cellular OR adolescen* OR youth* OR clinic* OR disease OR ecolog* OR geolog* OR financ* OR economic* OR *ergonomic* OR ecosystem* OR crime OR morpholog* OR anatom* OR school* OR war)

Biblioteca Digital Brasileira de Teses e Dissertações: (((model* OR UML OR Motorola) AND (Aspect* OR AOP OR POA)) OR "Design by Contract") NOT (mathemat* OR hospit* OR biolog* OR health* OR medic* OR *chemic* OR marketing OR food OR restaurant OR drink OR Meteorolog* OR Psycholog* OR electr* OR merchandising OR social OR temperature OR physics OR mechatronic* OR philosoph* OR alcohol OR mechanic* OR physiolog* OR Intertextuality OR acoustic OR pollution OR pedagog* OR "verb phrase"OR gis OR telecommunication OR cellular OR adolescen* OR youth* OR clinic* OR disease OR ecolog* OR geolog* OR financ* OR economic* OR *ergonomic* OR ecosystem* OR crime OR morpholog* OR anatom* OR school* OR war)

Filtro Ano de Publicação: 1997-2017

Filtro de Grau: Tese

EBSCOhost: (((model* OR UML OR Motorola) N3 (Aspect* OR AOP OR POA)) OR "Design by Contract") NOT (mathemat* OR hospit* OR biolog* OR health OR medic* OR *chemic* OR marketing OR food OR restaurant OR drink OR Meteorolog* OR Psycholog* OR electr* OR merchandising OR social OR temperature OR physics OR mechatronic* OR philosophy OR alcohol OR mechanic* OR physiolog* OR Intertextuality OR acoustic OR pollution OR pedagog* OR "verb phrase"OR gis OR telecommunication OR cellular OR adolescence)

Filtro Limitador: Texto Completo

Filtro Limitador: Referências Disponíveis

Filtro Limitador: 1997-2017

Filtro de Idioma: Inglês e Português

Google Scholar: Como o Google Scholar tem um sistema de busca próprio e que oferece um retorno muito amplo de resultados baseado nos algoritmos internos de relevância, modificou-se a estratégia para obter um resultado mais eficiente. Foram elaborados 3 (três) termos de busca simples, sendo que no primeiro, aproveitou-se 3 (três) páginas de retorno do Google Scholar e no segundo e terceiro, como corresponde a uma busca mais específica, aproveitou-se apenas a primeira página de retorno. Vale ressaltar que a Google Scholar sofre constante atualização em função das pesquisas dos usuários; sendo assim, é possível que em pouco tempo os resultados com os mesmos termos de busca sofram alterações, podendo gerar variações ao tentar reproduzir as buscas com esta ferramenta.

1º Termo de Busca: aspect uml

2º Termo de Busca: motorola aspect orient*

3º Termo de Busca: "Design by Contract"

Filtro de Período: 1997-2017

Filtro: Incluir citações

SciELO: (((model* OR UML OR Motorola) AND (Aspect* OR AOP OR POA)) OR "Design by Contract") NOT (mathemat* OR hospit* OR biolog* OR health OR medic* OR *chemic* OR marketing OR food OR restaurant OR drink OR Meteorolog* OR Psycholog* OR electr* OR merchandising OR social OR temperature OR physics OR mechatronic* OR philosophy OR alcohol OR mechanic* OR physiolog* OR Intertextuality OR acoustic OR pollution OR pedagog* OR "verb phrase" OR gis OR telecommunication OR cellular OR adolescence)

Filtro de Idioma: Inglês e Português

Filtro SciELO Áreas: Ciências Exatas e da Terra

Filtro SciELO Áreas: Engenharia

Filtro SciELO Áreas: Multidisciplinar

Filtro WoS Áreas: Engenharia Multidisciplinar

Filtro WoS Áreas: Ciências Multidisciplinares

Filtro WoS Áreas: Ciência da Computação: Engenharia de Software

Filtro WoS Áreas: Ciência da Computação: Aplicações Interdisciplinares

Filtro WoS Áreas: Ciência da Computação: Teoria e Métodos

SCOPUS: (((model* OR UML OR Motorola) N/3 (Aspect* OR AOP OR POA)) OR "Design by Contract") NOT (mathemat* OR hospit* OR biolog* OR health OR medic* OR *chemic* OR marketing OR food OR restaurant OR drink OR Meteorolog* OR Psycholog* OR electr* OR merchandising OR social OR temperature OR physics OR mechatronic* OR philosophy OR alcohol OR mechanic* OR physiolog* OR Intertextuality OR acoustic OR pollution OR pedagog* OR "verb phrase" OR gis OR telecommunication OR cellular OR adolescence)

Filtro Ano de Publicação: 1997-2017

Filtro de Área: Engenharia e Ciência da Computação

Como crivo para filtragem dos trabalhos levantados com a *string* de busca elaborada, optou-se por avaliar os seguintes elementos: Título, Resumo (*Abstract*) e Palavras-Chave (*Keywords*) levando em consideração os seguintes critérios para avaliar se o trabalho em questão deve ser incluído ou excluído na elaboração da revisão:

Critérios de Inclusão:

- Trabalho sobre Modelagem POA que estendam a UML;

- Trabalho sobre *Design by Contract*;
- Estudo de Caso sobre *Design by Contract*;
- Revisão Sistemática da Literatura sobre POA.

Critérios de Exclusão:

- Trabalho sobre Modelagem POA independente da UML;
- Trabalho sobre outras extensões da UML;
- Trabalho sobre outras extensões da POA;
- Revisão Sistemática da Literatura sobre extensões de POA;
- Estudo de Caso sobre UML.

Após a avaliação das publicações, faz-se necessário classificá-las para organizar e direcionar o estudo em relação as publicações mais relevantes. Como não há um padrão de classificação ideal para qualquer pesquisa, cabe ao pesquisador, formalizar um padrão de classificação que mais se adapte ao trabalho.

É importante avaliar a qualidade dos estudos primários para apoiar o processo de inclusão / exclusão e a alocação de ponderação para estudos específicos durante a fase de síntese de dados. Não há definições universais de qualidade de estudo, mas sugeriu-se que a qualidade se relaciona com a medida em que o viés é minimizado e que a validação externa e interna é maximizada. (BRERETON *et al.*, 2007 apud KHAN *et al.*, 2001, p. 579)⁶

Sob essa óptica, definiu-se um padrão de classificação simples, separando as relevâncias com os seguintes conceitos:

- Alta Relevância;
- Média Relevância;
- Baixa Relevância.

Para se catalogar os estudos selecionados foi desenvolvido um dicionário de dados a fim de se extrair os seus meta-dados, conforme o Quadro 11, que se encontra no Protocolo da Revisão Sistemática da Literatura no Apêndice B.

⁶ [There are no universal definitions of study quality but it has been suggested that quality relates to the extent to which bias is minimised and external and internal validation are maximised.]

O objetivo desta etapa é projetar formulários de extração de dados para registrar com precisão a informação que os pesquisadores obtêm dos estudos primários. Para reduzir a oportunidade de viés, formulários de extração de dados devem ser definidos e pilotados quando o protocolo do estudo é definido. (KITCHENHAM; CHARTERS, 2007, p. 29, tradução nossa)⁷

3.3 Documentação da Revisão

Na última etapa da Revisão Sistemática da Literatura organiza-se a divulgação por meio desta própria dissertação, ficando disponível na biblioteca da UFPE, e de seus repositórios digitais. A formatação deste trabalho respeita os padrões impostos pela ABNT e o modelo de avaliação da publicação é por meio de uma banca examinadora escolhida pela UFPE.

3.4 Resultados da Revisão Sistemática da Literatura

Nesta seção, será apresentada a evolução da construção do Resultado da Revisão Sistemática da Literatura, em busca dos trabalhos mais relevantes para a construção desta dissertação.

3.4.1 Fase 1: Levantamento Inicial da Literatura

Primeiramente, foram aplicados os termos de busca em cada fonte de dados, conforme descrito na Seção 3.2, cujos dados foram compilados na Tabela 1, que apresentam as quantidades de publicações de cada fonte de dados dentro do intervalo temporal definido para a Revisão Sistemática de Literatura, totalizando 288 (duzentos e oitenta e oito) publicações.

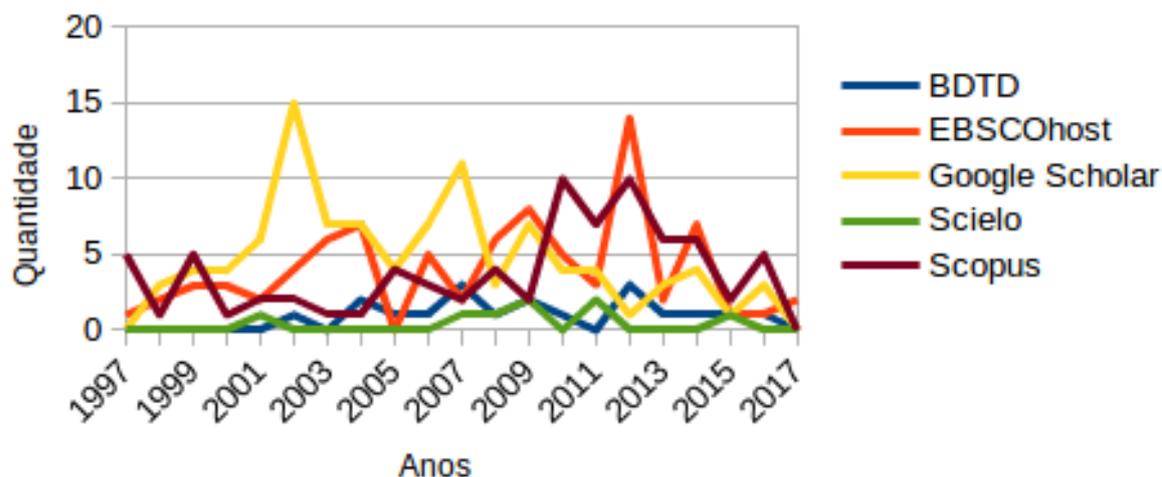
No Gráfico 1, está representada a evolução da quantidade de publicações relevantes por um gráfico em linha, conforme definido pelos termos de busca a esta dissertação ao longo do tempo, limitado pelo horizonte temporal definido previamente.

3.4.2 Fase 2: Avaliação da Literatura Segundo Critérios de Inclusão e Exclusão

Como primeiro fator de avaliação, visando descartar as publicações duplicadas, utilizou-se a ferramenta "Mendeley"⁸, para auxiliar nesta atividade. Como resultado, descartou-se 4 (quatro) publicações, por terem sido selecionados em duas fontes de buscas distintas, como pode ser visto na Tabela 2.

⁷ [The objective of this stage is to design data extraction forms to accurately record the information researchers obtain from the primary studies. To reduce the opportunity for bias, data extraction forms should be defined and piloted when the study protocol is defined.]

⁸ "Mendeley" é uma ferramenta de administração e controle de publicações científicas que auxilia o pesquisador na organização, guarda de documentos e citações.

Gráfico 1 – Publicações em Fontes de Dados x Horizonte Temporal

Fonte: Desenvolvido pelo autor.

Como segundo fator de avaliação, foi feita uma revisão dos títulos das 284 (duzentos e oitenta e quatro) publicações para descartar os trabalhos que não tem relação alguma com os objetivos desta dissertação. Apesar do termo de busca ter sido elaborado com uma camada excludente de palavras de outras áreas como saúde, geografia, artes dentre outras, ainda assim, várias publicações tiveram que ser descartadas por não ter relação com o tema em estudo.

Após a avaliação dos títulos das publicações, 188 (cento e oitenta e oito) trabalhos foram descartados, conforme pode ser visto na Tabela 3

Como terceiro fator de avaliação considerou-se os Resumos (*Abstract*) e Palavras-Chaves (*Keywords*), sendo que após a leitura dos fragmentos em observação das 96 (noventa e seis) publicações, descartou-se 25 (vinte e cinco) trabalhos científicos, por se distanciarem do tema principal desta dissertação, restando ainda 71 (sessenta e um) publicações para seguir com os critérios propostos de avaliação, conforme disposto na Tabela 4.

Após o último fator de avaliação empregado, descartou-se mais 3 (três) publicações devido ao seu conteúdo não contribuir com o desenvolvimento desta dissertação, conforme pode ser observado na Tabela 5.

Tabela 1 – Publicações em Fontes de Dados x Horizonte Temporal

Ano	BDTD	EBSCOhost	Google Scholar	SciELO	Scopus
1997	0	1	0	0	5
1998	0	2	3	0	1
1999	0	3	4	0	5
2000	0	3	4	0	1
2001	0	2	6	1	2
2002	1	4	15	0	2
2003	0	6	7	0	1
2004	2	7	7	0	1
2005	1	0	4	0	4
2006	1	5	7	0	3
2007	3	2	11	1	2
2008	1	6	3	1	4
2009	2	8	7	2	2
2010	1	5	4	0	10
2011	0	3	4	2	7
2012	3	14	1	0	10
2013	1	2	3	0	6
2014	1	7	4	0	6
2015	1	1	1	1	2
2016	1	1	3	0	5
2017	0	2	0	0	0
Total	19	84	98	8	79

Fonte: Desenvolvido pelo autor.

Tabela 2 – Descarte de Publicações Duplicadas

Estágio da Avaliação	Quantidade de Publicações
Publicações Levantadas	288
Publicações Duplicadas	(4)
Publicações Após Descarte	284

Fonte: Desenvolvido pelo autor.

Tabela 3 – Descarte de Publicações Após Avaliação de Títulos

Estágio da Avaliação	Quantidade de Publicações
Publicações Após Descarte dos Duplicados	284
Publicações Descartadas por Avaliação de Título	(188)
Publicações Após Descarte	96

Fonte: Desenvolvido pelo autor.

Tabela 4 – Descarte de Publicações Após Avaliação de Resumos e Palavras-Chave

Estágio da Avaliação	Quantidade de Publicações
Publicações Após Descarte da Avaliação de Título	96
Publicações Descartadas por Avaliação de Resumo e Palavra-Chave	(25)
Publicações Após Descarte	71

Fonte: Desenvolvido pelo autor.

Tabela 5 – Descarte de Publicações Após Leitura Completa do Trabalho

Estágio da Avaliação	Quantidade de Publicações
Publicações Após Descarte da Avaliação de Resumo e Palavra-Chave	71
Publicações Descartadas por Avaliação do Conteúdo	(3)
Publicações Após Descarte	68

Fonte: Desenvolvido pelo autor.

3.4.3 Fase 3: Classificação das Publicações Selecionadas

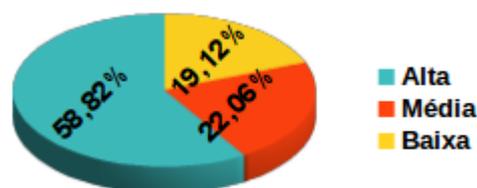
O processo de classificação das 68 (sessenta e oito) publicações selecionadas na fase anterior, após leitura, tiveram suas relevâncias avaliadas considerando a aderência ao estudo. Tal classificação apresentou como resultado, 40 (quarenta) publicações definidas como alta relevância, 15 (quinze) publicações definidas como média relevância e 13 (treze) publicações definidas como baixa relevância, conforme pode ser visto na Tabela 6 e no Gráfico 2.

Tabela 6 – Classificação Inicial das Publicações

Tipos de Classificação	Quantidade de Publicações
Alta	40
Média	15
Baixa	13
Total	68

Fonte: Desenvolvido pelo autor.

Gráfico 2 – Classificação Inicial das Publicações



Fonte: Desenvolvido pelo autor.

3.4.4 Fase 4: Busca de Literatura, Avaliação e Classificação com *Snowballing*

Segundo Greenhalgh e Peacock (2005), o método "*Snowballing*" é uma busca de referências ascendentes e descendentes das publicações pré-selecionadas. Este é um método importante para identificar fontes de alta qualidade, com uma relação direta com as publicações previamente selecionadas, e que por sua vez, não foi rastreado pela estratégia de busca.

Visto que o processamento da classificação já foi executado, decidiu-se aplicar a técnica *Snowballing* apenas nas publicações científicas classificadas como de alta relevância para a pesquisa, pois a aplicação do *Snowballing* das publicações de Média e Baixa Relevância, geraria como resultado um volume considerável de trabalhos com relevância igual ou inferior as suas publicações ascendentes, o que não acrescentaria uma colaboração substancial a Revisão Sistemática da Literatura.

O *Snowballing* foi aplicado de forma ascendente, que vem a ser uma pesquisa sobre as referências citadas nessas publicações, e descendente, que vem a ser o levantamento das publicações que citaram esses trabalhos de alta relevância e foi coletado um novo conjunto de publicações cujo teor tenha alguma relação com esta dissertação.

Esse novo conjunto de trabalhos também foi classificado e sobre as publicações de alta relevância para a pesquisa desse novo conjunto, executou-se um novo ciclo de

Snowballing. Foi necessária a repetição de 3 (três) ciclos de *snowballing* para se esgotar as publicações de alta relevância para esta pesquisa.

Para auxiliar nessa atividade, utilizou-se os recursos do sítio de busca "Google Scholar" que oferece a função de listar citações de uma determinada publicação.

Com resultado do *Snowballing* levantou-se inicialmente 68 (sessenta e oito) publicações, sendo que, após a aplicação dos mesmos fatores de avaliação descritos anteriormente, reduziu-se à 26 (vinte e seis) novas publicações inseridas ao contexto da Revisão Sistemática da Literatura conforme Tabela 7.

Tabela 7 – *Snowballing*: Avaliação das Publicações

Fase da Avaliação do <i>Snowballing</i>	Quantidade de Publicações
Levantamento Inicial do <i>Snowballing</i>	69
Descarte por Duplicidade	(16)
Descarte por Avaliação de Título	(10)
Descarte por Avaliação de Resumo	(15)
Descarte por Avaliação de Conteúdo	(2)
Total de Publicações Incluídas por <i>Snowballing</i>	26

Fonte: Desenvolvido pelo autor.

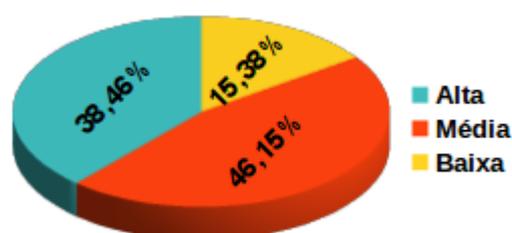
O processo de classificação também se estendeu as publicações selecionadas pelo *Snowballing*. Após leitura das publicações selecionadas, tiveram suas relevâncias avaliadas considerando a aderência ao estudo. Tal classificação definiu 10 (dez) publicações como alta relevância, 12 (doze) publicações como média relevância e 2 (duas) publicações como baixa relevância, conforme a Tabela 8 e o Gráfico 3.

Tabela 8 – Classificação das Publicações do *Snowballing*

Tipos de Classificação	Quantidade de Publicações
Alta	10
Média	12
Baixa	4
Total	26

Fonte: Desenvolvido pelo autor.

Gráfico 3 – Classificação das Publicações do *Snowballing*



Fonte: Desenvolvido pelo autor.

3.4.5 Fase 5: Resumo da Revisão Sistemática de Literatura

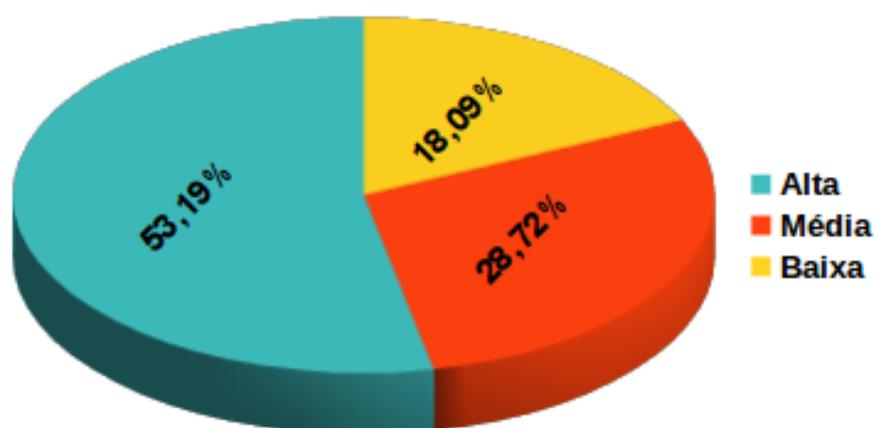
Somando-se as publicações selecionadas no "Levantamento Inicial de Literatura" e as publicações selecionadas no "Snowballing", resultou em 94 (noventa e quatro) trabalhos, cuja classificação por relevância foi definida como 50 (cinquenta) publicações com alta relevância, 27 (vinte e sete) publicações com média relevância e 17 (dezesete) publicações com baixa relevância, conforme descrito na Tabela 9 e no Gráfico 4

Tabela 9 – Classificação Final das Publicações

Tipos de Classificação	Quantidade de Publicações		
	Levantamento Inicial	<i>Snowballing</i>	Total
Alta	40	10	50
Média	15	12	27
Baixa	13	4	17
Total	68	26	94

Fonte: Desenvolvido pelo autor.

Gráfico 4 – Classificação Final das Publicações



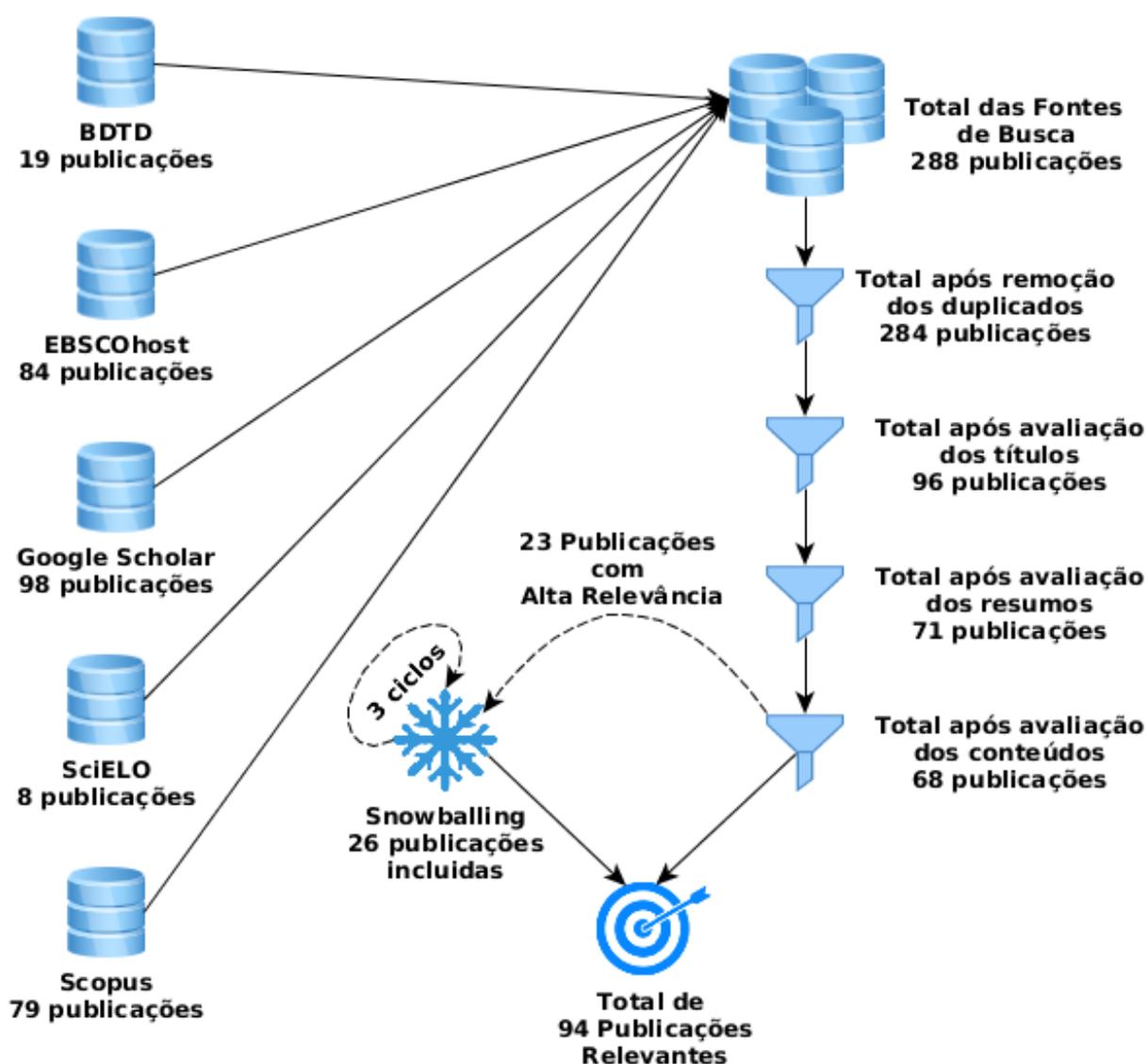
Fonte: Desenvolvido pelo autor.

Na figura 6 está representado graficamente o resumo de todo o processo para a seleção das 94 (noventa e quatro) publicações científicas resultantes desta Revisão Sistemática da Literatura, baseado na metodologia descrita por Kitchenham e Charters (2007), cujo objetivo foi levantar as publicações que versam sobre o tema de modelagem de aspectos.

Na representação gráfica estão pontuadas todas as fases da execução da busca e seleção, como se segue:

- Fontes de busca com os referidos valores nominais;
- Todos os passos do processo de seleção com o seu fluxo e valores;
- *Snowballing* com a descrição dos ciclos e valores;
- Resultado da Revisão Sistemática da Literatura sobre POA.

Figura 6 – Resumo da Revisão Sistemática da Literatura

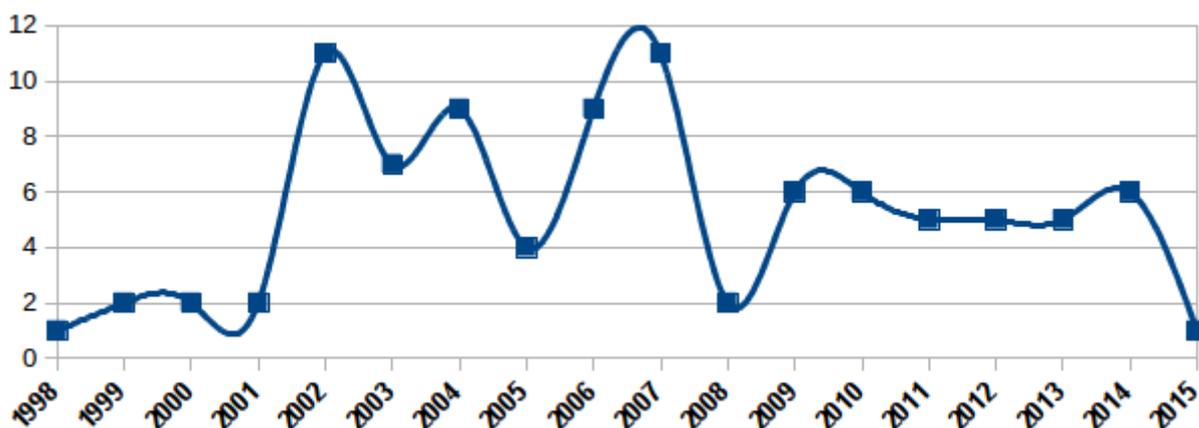


Fonte: Desenvolvido pelo autor.

Ao consolidar as publicações oriundas do levantamento inicial de publicações pelas fontes de buscas e do *Snowballing* e redimensionar a distribuição dessas publicações no

horizonte temporal, gerou-se o Gráfico 5 e é possível perceber que a maior incidência de trabalhos científicos selecionados para esta pesquisa estão nos anos de 2002 (dois mil e dois) e 2007 (dois mil e sete). Esse fato ocorreu porque no ano de 2002 (dois mil e dois), durante a conferência *Conference on Aspect-Oriented Software Development (AOSD)*, ocorreu o *Workshop on Aspect-Oriented Modeling With UML (AOM)*, que teve uma quantidade expressiva de trabalhos científicos relevantes para esta dissertação e no ano de 2007 (dois mil e sete) o periódico *Journal of Object Technology* também publicou uma série de artigos igualmente significativos.

Gráfico 5 – Publicações x Horizonte Temporal



Fonte: Desenvolvido pelo autor.

A lista completa das 94 (noventa e quatro) publicações selecionadas está descrita no Apêndice C, no formato já estabelecido no Protocolo da Revisão Sistemática da Literatura, conforme a Seção 11. E a lista completa de todos os periódicos e meios de publicações destas publicações encontra-se no Apêndice D.

Após a conclusão do processo de seleção e classificação das publicações, executou-se o processo de categorização desses trabalhos, conforme Tabela 10 e o Gráfico 6. As publicações selecionadas foram categorizadas em 11 (onze) grupos, dentre eles 5 (cinco) técnicas de extensão para a UML, outras Revisões Sistemáticas de Literatura sobre Modelagem de Aspectos, AOM, Geração de Código, Implementações, a Técnica *Design by Contract* e *Design by Contract* sobre AOP.

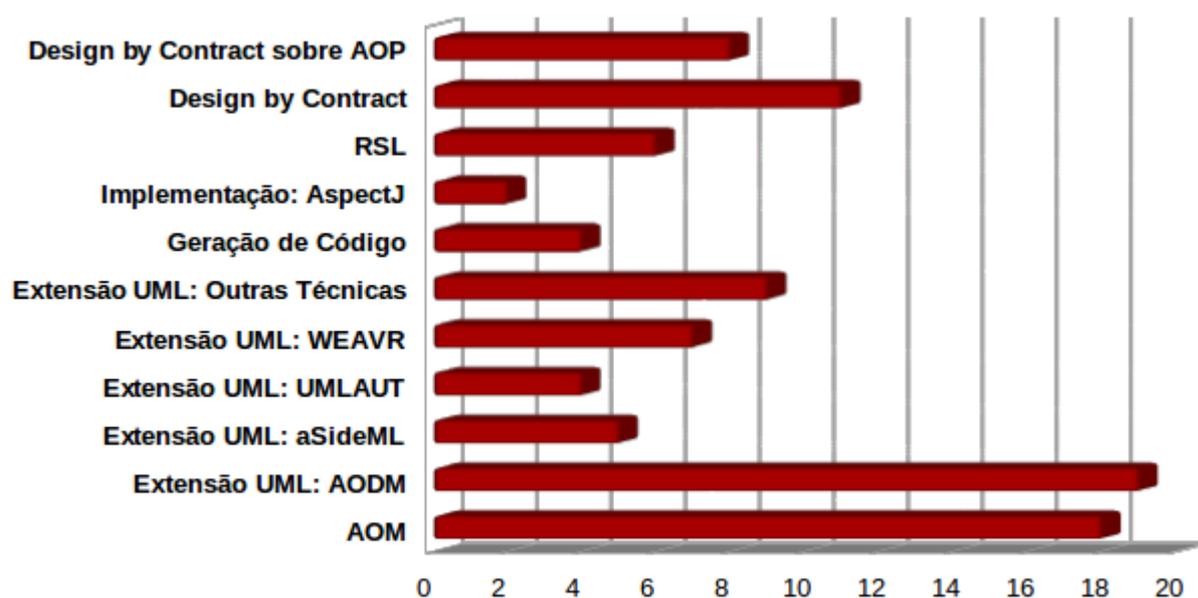
3.5 Síntese dos Resultados

Para concluir a Revisão Sistemática da Literatura optou-se pela elaboração da síntese dos dados levantados, com o mapeamento e a avaliação do conteúdo dos trabalhos publicados utilizando a técnica da "triangulação ecológica" que, segundo Banning (2001),

Tabela 10 – Categorização das Publicações

Categoria	Quantidade de Publicações
AOM	18
Extensão UML: AODM	19
Extensão UML: aSideML	5
Extensão UML: UMLAUT	7
Extensão UML: WEAVR	7
Extensão UML: Outras Técnicas	9
Geração de Código	4
Implementação: AspectJ	2
RSL	6
<i>Design by Contract</i>	12
<i>Design by Contract sobre AOP</i>	8
Total	94

Fonte: Desenvolvido pelo autor.

Gráfico 6 – Categorização das Publicações

Fonte: Desenvolvido pelo autor.

sua essência é a interdependência entre teoria, método e resultados das técnicas levantadas pela RSL. O resultado desta técnica de síntese produz informações sobre quais trabalhos são mais adequados para se atingir melhores resultados.

Com a categorização da publicações, que pode ser observada no Gráfico 6, propiciou-se a avaliação do conteúdo das técnicas levantadas e baseado no trabalho de Losavio, Matteo e Morantes (2009), que compara as técnicas de modelagem de aspectos, pode-se

tabular os trabalhos publicados na área. "O processo de síntese pode iniciar-se com uma tabulação dos estudos primários selecionados, de forma a construir um mapa dos resultados obtidos." (DRESCH; LACERDA; ANTUNES JÚNIOR, 2015, p. 168)

3.5.1 Síntese dos Resultados de Modelagem de Aspectos

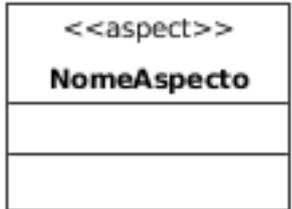
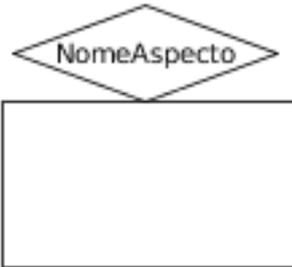
Nos Quadros 6 e 7 pode-se comparar a aplicação de modelo dos componentes do paradigma de Orientação a Aspectos das diversas publicações resultantes da categorização da RSL. Para a elaboração da comparação entre as técnicas foram considerados os trabalhos com representações próprias para o Aspecto (Quadro 6) ou seus componentes (Quadro 7).

As técnicas de modelagem propostas em Barra, Génova e Llorens (2004) e Zakaria, Hosny e Zeid (2002) não possuem um nome específico. Portanto, serão representadas pelos autores da publicação. As técnicas "*Composition Pattern*", "UMLAUT" e "WEAVR", não possuem representação própria para Aspectos, ou seja, os Aspectos não são de primeira classe, são representações de classes simples, em uma estrutura montada no Diagrama de Classe que representa o funcionamento do aspecto; portanto, no Quadro 7 no qual observa-se os demais componentes que compõe o Aspecto, essas técnicas não serão representadas.

Com os dados tabulados pôde-se efetuar a triangulação entre o problema x teoria (trabalhos estudados) x avaliação ou resultado, conforme exposto no Quadro 8.

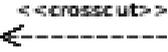
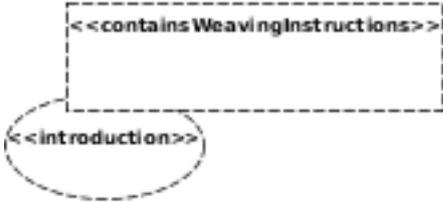
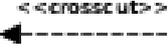
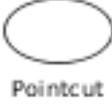
Para se avaliar os trabalhos resultantes da Revisão Sistemática da Literatura, basicamente foram avaliados 2 (dois) fatores: a expressividade dos componentes e seu comportamento no Diagrama de Classe Estendido. Ao avaliar o primeiro fator, somente a AODM e a aSideML comportaram todos os componentes de Aspectos esperado. Ao avaliar o segundo fator percebeu-se que a aSideML, necessita de um diagrama complementar ao Diagrama de Classe para representar a relação entre Aspectos e Classes. Como o propósito da dissertação é representar Design by Contract sobre Classes e Aspectos em um Diagrama de Classe, o único trabalho que respondeu positivamente ao estudo foi o AODM; sendo assim, essa foi a técnica escolhida para o desenvolvimento da pesquisa.

Quadro 6 – Síntese dos Dados: Aspectos

TÉCNICA	PUBLICAÇÃO	ASPECTO	ARTEFATO
AO-UML	Wang <i>et al.</i> (2006)		Diagrama de Classe Estendido (Próprio)
AODM	Aldawud, Elrad e Bader (2001) e Stein, Hanenberg e Unland (2002b)		Diagrama de Classe Estendido (Próprio)
aSideML	Chavez (2004)		Diagrama de Classe Estendido (Próprio) e Diagramas de Aspecto
-	Barra, Génova e Llorens (2004)		Diagrama de Classe Estendido (Próprio)
<i>Composition Pattern</i>	Clarke e Walker (2001)	Não há representação própria	Diagrama de Classe
UMLAUT	Ho, Pennaneac'H e Plouzeau (2000)	Não há representação própria	Diagrama de Classe
WEAVR	Cottenier, Van Den Berg e Elrad (2006)	Não há representação própria	Diagrama de Classe
-	Zakaria, Hosny e Zeid (2002)		Diagrama de Classe Estendido (Próprio)

Fonte: Desenvolvido pelo autor.

Quadro 7 – Síntese dos Dados: Componentes do Aspecto

TÉCNICA	COMPONENTES			
	CROSSCUT	JOIN POINT	ADVICE	POINTCUT
AO-UML		-	-	-
AODM			<<advice>>	<<pointcut>>
aSideML		<<NomeJoinPoint>>	_xxx (before) xxx_ (after) _xxx_ (around)	
Modelagem de Barra, Génova e Llorens (2004)			-	
Modelagem de Zakaria, Hosny e Zeid (2002)		-	<<before>> <<after>> <<around>> and others	

Fonte: Desenvolvido pelo autor.

Quadro 8 – Matriz da Síntese dos Dados sobre AOM

TÉCNICA	COMPONENTES	ARTEFATO	RESULTADO
AO-UML	Incompleto	Diagrama de Classe Estendido (Próprio)	Negativo
AODM	Completo	Diagrama de Classe Estendido (Próprio)	Positivo
aSideML	Completo	Diagrama de Classe Estendido (Próprio) e Diagramas de Aspecto	Negativo
Modelagem de Barra, Génova e Llorens (2004)	Incompleto	Diagrama de Classe Estendido (Próprio)	Negativo
<i>Composition Pattern</i>	Incompleto	Diagrama de Classe	Negativo
UMLAUT	Incompleto	Diagrama de Classe	Negativo
WEAVR	Incompleto	Diagrama de Classe	Negativo
Modelagem de Zakaria, Hosny e Zeid (2002)	Incompleto	Diagrama de Classe Estendido (Próprio)	Negativo

Fonte: Desenvolvido pelo autor.

3.5.2 Síntese dos Resultados de Design by Contract sobre Aspectos

A Revisão Sistemática da Literatura apresentou 7 (sete) trabalhos como resultado da busca por aplicações de DbC sobre Aspectos. A avaliação desses trabalhos se deu com a observação das bibliotecas (libs) de execução de AOP e DbC, conforme pode ser observado no Quadro 9.

A Classificação do Resultado foi positiva em contrapartida às demais técnicas, porque a Lib DbC é baseada no JML que é uma das implementações de DbC de maior notoriedade no mercado de desenvolvimento de software com uma comunidade de desenvolvimento muito ativa e de muita expressividade. Sob esse critério, o AspectJML e o Pipa se apresentam como técnicas pautadas na JML.

Outra razão para a adoção da AspectJML no desenvolvimento deste trabalho se respalda no segundo critério de avaliação (Lib AOP), que utiliza o @AspectJ, que vem a ser uma implementação AspectJ baseada em anotações, ou *annotations*.

Quadro 9 – Matriz da Síntese dos Dados: *Design by Contract* sobre AOP

TÉCNICA	Lib Dbc	Lib AOP	RESULTADO
AspectJML	Próprio baseado em JML	@AspectJ	Positivo
Barter	Próprio (descontinuado)	AspectJ	Negativo
ConaJ	Próprio	AspectJ	Negativo
Contract4J	JavaBeans (ContractBeans)	AspectJ	Negativo
ContractAJ	Próprio	AspectJ	Negativo
Jose	Próprio	AspectJ	Negativo
Pipa	JML	AspectJ	Negativo

Fonte: Desenvolvido pelo autor.

3.6 Critérios de Documentação da RSL

A Documentação da Revisão Sistemática da Literatura e a disseminação dos seus resultados será por meio da própria dissertação que motivou o trabalho de revisão, ficando disponível na biblioteca da UFPE, e de seus repositórios digitais, tendo sua formatação de acordo com os padrões impostos pela ABNT e a sua avaliação se dará por meio de uma banca examinadora escolhida pela UFPE.

4 ABORDAGEM DAS TÉCNICAS BASE

Para o desenvolvimento do Projeto do Artefato desta dissertação, segue-se neste capítulo a descrição das técnicas levantadas na Revisão Sistemática da Literatura, que servirão de base para o Projeto do Artefato. Essas técnicas são AODM para servir de base para o Diagrama de Classes com Aspectos, e a sintaxe do AspectJML para se expressar os exemplos que ilustram a apresentação do Projeto do Artefato.

4.1 Java

Segundo (DEITEL; DEITEL, 2005, p. 6) a linguagem Java, começou a ser desenvolvida pela empresa Sun Microsystems em 1991, sob o codinome Green, e em 1995 foi formalmente anunciada chamando a atenção da comunidade de desenvolvimento de software.

Para se desenvolver utilizando a linguagem Java necessita-se de um pacote JDK (*Java Development Kit*¹, ou Pacote para Desenvolvimento Java). Para executar um programa desenvolvido pela linguagem Java é necessário ter no computador o pacote JRE (*Java Runtime Environment*², ou Ambiente de Tempo de Execução Java) e a JVM (*Java Virtual Machine*³, ou Máquina Virtual Java).

4.2 AspectJ

AspectJ é uma implementação de orientação a aspectos para a linguagem Java de grande notoriedade no mercado de desenvolvimento de software. A compilação do AspectJ é feita através de *byte code* Java padrão e sua implementação, manutenção e legibilidade do código são relativamente simples, o que facilitou sua adoção pelo mercado.

AspectJ é uma extensão orientada a aspectos simples e prática para Java. Com apenas algumas construções novas, o AspectJ fornece suporte para a implementação modular de uma série de *crosscutting concerns*. Os *join points* são os principais pontos na execução do programa; os pontos são coleções de *join points*; *advice* é um método especial como a construção que pode ser anexado a pontos de referência; e os aspectos são unidades modulares de implementação de *crosscutting*, constituídas por pontos, conselhos e declarações comuns de membros Java. O código AspectJ é compilado em *bytecode* Java padrão. Extensões simples aos ambientes de desenvolvimento Java existentes permitem navegar na estrutura

¹ [Em função do termo *Java Development Kit* ser muito difundido na literatura técnica brasileira, usaremos o termo em inglês, bem como sua sigla, no decorrer desta dissertação]

² [Em função do termo *Java Runtime Environment* ser muito difundido na literatura técnica brasileira, usaremos o termo em inglês, bem como sua sigla, no decorrer desta dissertação]

³ [Em função do termo *Java Virtual Machine* ser muito difundido na literatura técnica brasileira, usaremos o termo em inglês, bem como sua sigla, no decorrer desta dissertação]

de *crosscutting* de aspectos da mesma maneira que se navega na estrutura de herança das classes. Vários exemplos mostram que o AspectJ é poderoso e que os programas que o usam na codificação são fáceis de entender.(KICZALES *et al.*, 2001, p. 1, tradução nossa)⁴

Conforme *Ibid.*, p. 2 o AspectJ preza pela compatibilidade com códigos Java atendendo a 4 (quatro) premissas:

- Compatibilidade Ascendente: Compatibilidade integral com programas escritos em Java;
- Compatibilidade de Plataforma: Compatibilidade com a *Java Virtual Machine*;
- Compatibilidade de Ferramenta: As plataformas de desenvolvimento Java devem dar suporte ao AspectJ;
- Compatibilidade com o Desenvolvedor: Para o desenvolvedor Java, a implementação do AspectJ deve ser feita de forma natural;

Para exemplificar a sintaxe do AspectJ, segue abaixo a classe Foo (escrita em linguagem Java) que recebe por parâmetro um texto e imprime em tela, seguido pelo aspecto AspectFoo que insere um texto fixo após a impressão do texto impresso pela classe Foo.

Arquivo "Foo.java":

```
1 public class Foo {
2
3     public static void texto(String texto) {
4         System.out.println(texto);
5     }
6
7 }
```

Arquivo "AspectFoo.aj":

⁴ [AspectJ is a simple and practical aspect-oriented extension to Java. With just a few new constructs, AspectJ provides support for modular implementation of a range of crosscutting concerns. Join points are principled points in the execution of the program; pointcuts are collections of join points; advice is a special method-like construct that can be attached to pointcuts; and aspects are modular units of crosscutting implementation, comprised of pointcuts, advice, and ordinary Java member declarations. AspectJ code is compiled into standard Java bytecode. Simple extensions to existing Java development environments make it possible to browse the crosscutting structure of aspects in the same kind of way as one browses the inheritance structure of classes. Several examples show that AspectJ is powerful, and that programs written using it are easy to understand.]

```
1 public aspect AspectoFoo {
2
3     pointcut confirmaTexto() : call (* Foo.texto(..));
4
5     after() : confirmaTexto() {
6         System.out.println("OK!");
7     }
8
9 }
```

4.3 AODM

A AODM ou *Aspect Oriented Design Model* foi a extenparametrosão da UML para modelagem de aspectos resultante da RSL para servir de base para o Projeto do Artefato desta dissertação. A técnica de se estender a UML para servir a novos propósitos se faz legítima, considerando a relevância da AOP para a área de desenvolvimento de software.

O UML suporta mecanismos de extensão, (estereótipos, valores e restrições marcados) para permitir que a UML se adapte às necessidades de um domínio específico. O perfil UML permite que as partes interessadas de um determinado domínio expressem a semântica de seus sistemas usando um conjunto bem definido de extensões. (ALDAWUD; ELRAD; BADER, 2001, p. 1, tradução nossa)⁵

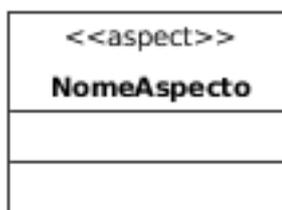
O modelo de *design* orientado a aspectos (AODM) é um novo modelo de design para o desenvolvimento de programas AspectJ com o UML. Estende os conceitos UML existentes usando mecanismos de extensão UML padrão para fornecer conceitos orientados a aspectos como eles são encontrados no AspectJ. Além disso, o AODM especifica como um modelo de design orientado a aspectos pode ser transformado em um modelo de projeto comum UML. Ele demonstra como esse mecanismo de tecelagem pode ser representado com a ajuda de conceitos UML existentes. O AODM facilita a percepção de *crosscutting* orientado a aspectos nos programas AspectJ e traz as vantagens da orientação de aspecto para o nível de *design*. (STEIN; HANENBERG; UNLAND, 2002b, p. 1, tradução nossa)⁶

⁵ [UML supports extension mechanisms, (stereotypes, tagged values, and constrains) to allow tailoring UML to fit the needs of a specific domain, UML profile allow the stakeholders of a certain domain to express the semantics of their systems using a well-defined set of extensions.]

⁶ [The Aspect-Oriented Design Model (AODM) is a new design model for the development of AspectJ programs with the UML. It extends existing UML concepts using standard UML extension mechanisms to provide aspect-oriented concepts as they are found in AspectJ. Further, the AODM specifies how an aspect-oriented design model may be transformed into an ordinary UML design model. It demonstrates how this weaving mechanism may be represented with help of existing UML concepts. The AODM facilitates the perception of aspect-oriented crosscutting in AspectJ programs and carries over the advantages of aspect-orientation to the design level. This paper explains the reasoning behind the AODM and demonstrates how it may be used to design aspect-oriented crosscutting in general.]

Conforme Ibid., p. 1, a AODM expressa aspectos com a representação gráfica de uma classe em UML adicionada do estereótipo <<aspect>>, conforme Figura 7

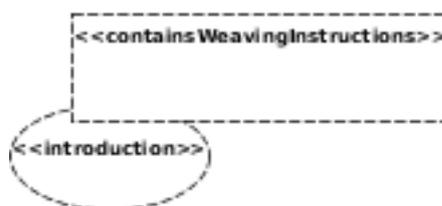
Figura 7 – Representação de um Aspecto conforme AODM



Fonte: Desenvolvido pelo autor.

Em relação aos componentes que envolvem o Aspecto, o *Advice* e o *Pointcut* são representados pelos estereótipos, <<advice>> e <<pointcut>> respectivamente. O *Crosscut* é representado por uma seta tracejada para efetuar a ligação e sinalizada com o estereótipo <<crosscut>>. O *Join Point* é representado por um elemento próprio, que é inserido dentro da representação do Aspecto, nele é descrita o alvo de implementação, conforme descrito em Stein, Hanenberg e Unland (2002a) e apresentado na Figura 8.

Figura 8 – Representação de um *Join Point* conforme AODM



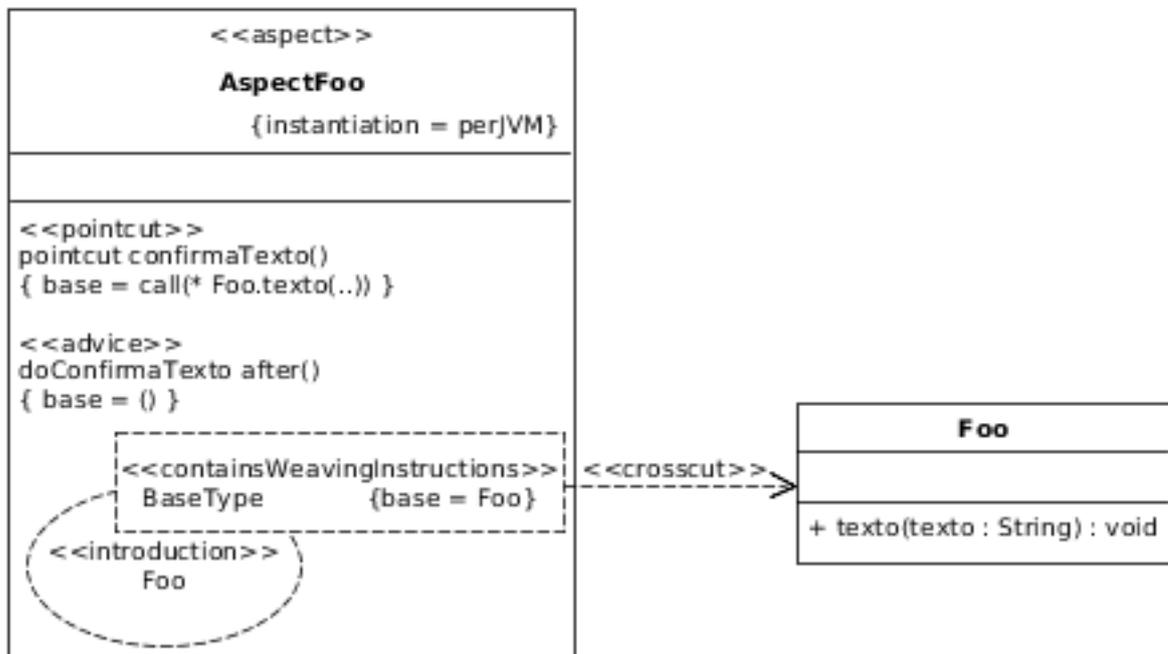
Fonte: Desenvolvido pelo autor.

Segue na Figura 9 com um exemplo prático de um modelo AODM derivado do código Java/AspectJ exposto na Seção 4.2.

4.4 Design by Contract

A *Design by Contract* aplica uma técnica que estabelece regras bem definidas para se validar o método em relação à passagem de parâmetros. Essas regras são chamadas

Figura 9 – Exemplo de Modelo AODM



Fonte: Desenvolvido pelo autor.

de "contratos" pela DbC e se comporta como um acordo formal na assinatura de métodos, sendo aplicadas em cláusulas de pré-condições e pós-condições.

Segundo (MEYER, 2000, p. 340), uma pré-condição expressa as restrições sob as quais uma rotina funcionará corretamente e uma pós-condição expressa propriedades do estado resultantes da execução de uma rotina.

O conceito-chave será Design by Contract: ver a relação entre uma classe e seus clientes como um acordo formal, expressando os direitos e obrigações de cada partido. Somente através de uma definição tão precisa de reivindicações e responsabilidades de cada módulo, podemos esperar alcançar um grau significativo de confiança nos grandes sistemas de software. Ibid., p. 331, tradução nossa⁷

4.5 JML

A JML é uma implementação para *Design by Contract* na linguagem Java, cujo repositório eletrônico do projeto encontra-se em Leavens (1999). Os contratos de DbC estabelecidos na sintaxe JML são declarados por um sinal de comentário seguido de um caractere arroba ou entre bloco de comentário e um caractere arroba.

⁷ [The key concept will be Design by Contract: viewing the relationship between a class and its clients as a formal agreement, expressing each party's rights and obligations. Only through such a precise definition of every module's claims and responsibilities can we hope to attain a significant degree of trust in large software systems. In]

As especificações JML são escritas em comentários de anotação especiais, que começam com um sinal de arroba (@). Atendemos ao início das linhas em comentários de anotação da forma `/*@ ... @*/`. Observe que o arroba, @, deve estar ao lado do início dos caracteres de comentário. Um comentário iniciando `// @` será ignorado pela JML, enquanto que `//@` inicia uma anotação corretamente. Da mesma forma, `/* @` não iniciará uma anotação, em vez disso, é necessário usar `/*@`. (As ferramentas JML atualmente não alertam sobre comentários que possam usar esses marcadores de anotação equivocados). (LEAVENS; CHEON, 2006, p. 1, tradução nossa)⁸

Seguindo a notação de início da escrita do contrato vêm as cláusulas e as condições. Como exemplo, o caso da cláusula de pré-condição *requires*, requer que o parâmetro recebido pelo método atenda a uma condição especificada no contrato e a cláusula de pós-condição *ensures* que garante que a saída do método atenda a uma condição especificada no contrato, dentre outras cláusulas suportadas.

No exemplo descrito abaixo o contrato DbC estabelece que a "variavel_x" tenha que ser maior que 0 (zero).

```
//@ requires variavel_x > 0;
```

4.6 AspectJML

Segundo Rebêlo *et al.* (2014b, p. 59), AspectJML é uma extensão orientada a aspectos simples e prática para o JML, e seu repositório encontra-se em Rebêlo *et al.* (2014a). Ele suporta a especificação de contratos de *crosscutting concerns* para código Java de forma modular. O AspectJML usa a sintaxe baseada em anotações como na implementação do AspectJ (também chamado de `@AspectJ`); a implementação da sintaxe baseado em anotações também está presente em outras linguagens e *frameworks* mais robustos, como o Spring Framework (2002), descrito em Johnson *et al.* (2016, p. 215), atualmente mantido pela empresa Pivotal Software (2013).

O AspectJML implementa contratos de DbC em aspectos baseado em como a técnica JML implementa contratos DbC em classes java, utilizando a notação de comentário (`//`), seguido da notação de *annotation* (`@`), resultando na notação `//@`.

No exemplo descrito abaixo o contrato DbC estabelece que o retorno do método "getNumero" da classe Foo seja obrigatoriamente um valor entre 0 (zero) e 10 (dez), inclusive ambos os extremos.

```
//@ requires foo.getNumero() >= 0 && foo.getNumero() <= 10;
```

⁸ [JML specifications are written in special annotation comments, which start with an at-sign (@). At-signs at the beginnings of lines in annotation comments of the form `/*@ ... @*/` are ignored. Note that the at-sign, @, must be right next to the start of comment characters. A comment starting `// @` will be ignored by JML, whereas `//@` starts an annotation properly. Similarly, `/* @` will not start an annotation, instead one must use `/*@`. (JML tools do not currently warn about comments that might use such mistaken annotation markers.)]

5 PROJETO DO ARTEFATO

Neste capítulo apresenta-se o Projeto do Artefato que trata o problema observado na Seção 1.3. O Artefato refere-se a uma extensão de um Diagrama de Classe já estendido para Aspectos para tratar *Design by Contract*. Para tal, o Projeto do Artefato utiliza-se da técnica exposta no *Aspect-Oriented Design Model* (AODM) para abrigar os contratos da DbC tanto nas Classes quanto nos Aspectos.

5.1 Desenvolvimento do Artefato

Em Cepa e Kloppenburg (2005), é desenvolvido uma possibilidade de representar *annotations* em classes modeladas em Diagramas de Classes. Nesta pesquisa, são avaliadas 6 (seis) possibilidades diferentes de se modelar *annotations*:

- Como Propriedade;
- Como Estereótipos;
- Como Modelo;
- Como Partição;
- Como Classes Separadas;
- Como Comentário;

Apesar de classificar cada modelo através de critérios pré-estabelecidos, como no Quadro 10, em que a classificação varia de - "(menor classificação) a "+ "(maior classificação), Cepa e Kloppenburg (2005) afirma que nenhuma representação é mais adequada do que a outra em relação a todos os critérios e que dependendo do peso relativo dos atributos no projeto, seus parâmetros e a densidade do uso de atributo em uma classe ou nos elementos internos da classe, diferentes notações podem ser usadas em diferentes situações.

Baseado no trabalho de Cepa e Kloppenburg (2005), para se representar em AODM os contratos DbC em classes e aspectos, optou-se pelo método de apresentação através de estereótipos, visto que a compatibilidade com o padrão UML é um dos fatores mais importantes, juntamente com a visibilidade.

Quadro 10 – Resumo das várias apresentações de atributos explícitos no UML

Apresentação	Padrão UML	Visibilidade	Clareza	Reusabilidade	Estrutura
Propriedade	++	--	--	-	-
Estereótipo	++	+	-	-	-
Modelo	+	+	-	-	+
Partição	-	+	-	-	+
Classe Separada	+	++	+	++	++
Comentário	+	+	-	++	--

Fonte: Baseado em Cepa e Kloppenburg (2005).

A avaliação de alguns fatores como clareza, reusabilidade e estrutura (respectivamente a quarta, quinta e sexta colunas do Quadro 10) apontam que a opção da representação por estereótipo pode não ser a mais adequada, porém, o fator clareza, refere-se a verbosidade do artefato, quanto menor a necessidade de escrita, mais claro torna-se o artefato; porém, como o contrato é uma condição muito particular do método, há a necessidade de uma descrição, logo esse critério, juntamente com o da reusabilidade não são tão importantes.

Quão clara ou detalhada é a notação. As classificações menos verbosas são preferidas. As notações verbais podem ser difíceis de gerenciar quando são usadas para decorar elementos de classe internos, por exemplo, parâmetros do método. As notações verbais também podem resultar em confusão do diagrama UML com muitos elementos. (CEPA; KLOPPENBURG, 2005, p. 3, tradução nossa)¹

Para simplificar a implementação não há a necessidade de uma estrutura própria para se representar o contrato. O objetivo é conseguir representar os contratos DbC em classes e aspectos através de um único artefato, o Diagrama de Classes AODM estendido.

As formas de apresentação "Classe Separada" e "Comentário", apesar de possuírem as melhores pontuações, segundo o Quadro 10, foram descartadas, pois seus usos não fazem sentido no caso do DbC, visto que os contratos são validadores do método (no caso das classes) e do advice (no caso dos aspectos), sendo assim, não seria expressivo destacar sua representação da classe ou aspecto. Logo, a representação por estereótipo se apresentou como mais adequada para se adaptar para contratos DbC.

No caso de se utilizar estereótipos para representar DbC, a cláusula do contrato é usado como um estereótipo para decorar elementos UML. Para distinguir um estereótipo que

¹ [How clear or verbose is the notation. The less verbose notations are preferred. Verbose notations can be difficult to manage when they are used to decorate internal class elements, e.g., method parameters. Verbose notations could also result in cluttering of the UML diagram with too many elements.]

serve como cláusula do contrato de outro estereótipo, o conjunto de caracteres especiais estabelecidos pelo JML '@' deve ser adicionado no estereótipo '<<@ ... >>'. Uma extensão seria necessária para incluir a lista de condições da cláusula do contrato como uma notação especial sob forma de propriedade UML '{ ... }' seguindo o estereótipo.

Segue o formato padrão para expressar contratos DbC em Diagramas de Classe AODM:

```
<<@ cláusula>> {condição}
```

Para exemplificar a implementação do contrato em modelos representado por estereótipos segue o contrato estabelecido para o método depositar da classe ContaCorrente:

```
//@ requires valor > 0;
```

Segue a representação em modelo da transposição do código do contrato:

```
<<@ requires>> {valor > 0}
```

5.2 Exemplificação do Artefato

Para ilustrar a aplicação da extensão do artefato, bem como a evolução conceitual da questão abordada pela dissertação segue um exemplo didático de uma fração de um sistema bancário com uma Classe de Conta Corrente e um Aspecto que se propõe a gerar um *log* das ações resultantes de seus métodos.

Roteiro do Exemplo:

- **Parte 1:** Apresentação do código Java/AspectJ;
- **Parte 2:** Modelagem AODM do código apresentado;
- **Parte 3:** Transformação das condições (*if*) da Classe em contratos DbC utilizando JML;
- **Parte 4:** Transformação da condição (*if*) do Aspecto em Contratos utilizando AspectJML;
- **Parte 5:** Modelagem AODM estendida para DbC.

5.2.1 Exemplo (Parte 1): Código Java/AspectJ

Segue um pequeno exemplo de um código Java/AspectJ de uma classe simples "ContaCorrente" onde é possível efetuar saque, depósito e consulta de saldo e um Aspecto que escreve *logs* dos depósitos e saques com sucesso.

Arquivo "ContaCorrente.java":

```
1 public class ContaCorrente {
2
3     private String numero;
4     protected Double saldo;
5
6     public String getNumero() {
7         return this.numero;
8     }
9
10    public Double getSaldo() {
11        return this.saldo;
12    }
13
14    public void criaContaCorrente(String numero) {
15        this.numero = numero;
16        this.saldo = 0.0;
17    }
18
19    public void depositar(Double valor) {
20
21        if(valor > 0) {
22            this.saldo = this.saldo + valor;
23        }
24
25    }
26
27    public void sacar(Double valor) {
28
```

```
29         if(this.saldo >= valor && valor > 0) {
30             this.saldo = this.saldo - valor;
31         } else {
32             System.out.println("Transação Cancelada: "
33                 + "Saldo insuficiente!");
34         }
35
36     }
37
38     public void consultaSaldo() {
39         System.out.println("Saldo - Conta Corrente: "
40             + this.numero + " - R$ " + this.saldo);
41     }
42
43 }
```

Arquivo "AspectoLogContaCorrente.aj":

```
1 import java.text.DateFormat;
2 import java.text.DecimalFormat;
3 import java.text.SimpleDateFormat;
4 import java.util.Date;
5
6 public aspect AspectoLogContaCorrente {
7
8     private DateFormat df = new
9         SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
10    private DecimalFormat formatter = new
11        DecimalFormat("#0.00");
12
13    pointcut logDeposito(ContaCorrente conta, Double valor) :
14        call(public void ContaCorrente.depositar(..)) &&
15        target(conta) && args(valor);
16
17    after(ContaCorrente conta, Double valor) :
18        logDeposito(conta, valor) {
19
```

```
20         System.out.println("[ " +
21             df.format((new Date()).getTime()) +
22             "] Depósito - Conta Corrente: " +
23             conta.getNumero() + - R$ " +
24             formatter.format(valor));
25     }
26
27     pointcut logSaque(ContaCorrente conta, Double valor) :
28         call(public void ContaCorrente.sacar(..)) &&
29         target(conta) && args(valor);
30
31     void around(ContaCorrente conta, Double valor) :
32         logSaque(conta, valor) {
33
34         if(conta.getSaldo() >= valor && valor > 0) {
35             System.out.println("[ "+
36                 df.format((new Date()).getTime()) +
37                 "] Saque - Conta Corrente: " +
38                 "conta.getNumero() + - (R$ " +
39                 formatter.format(valor).toString() + ")");
40         }
41         proceed(conta, valor);
42
43     }
44
45 }
```

5.2.2 Exemplo (Parte 2): AODM

Segue a modelagem dos códigos Java (Classe "ContaCorrente") e AspectJ (Aspecto "AspectoLogContaCorrente") ambos expostos na Subseção 5.2.1, conforme Figura 10 do Diagrama de Classes utilizando a técnica AODM.

Figura 10 – Exemplo de AODM



Fonte: Desenvolvido pelo autor.

5.2.3 Exemplo (Parte 3): JML

Avaliando a Classe `ContaCorrente` apresentada na Subseção 5.2.1, observa-se que há uma condição (*if*) no método "depositar" e outra condição (*if*) no método "sacar" que podem ser substituídos por contratos DbC.

Abaixo, segue a condição expressa no método "depositar":

```
19         if(valor > 0) {  
                ...  
21     }
```

Esta condição pode ser substituída pela assinatura do seguinte contrato:

```
//@ requires valor > 0;
```

Abaixo, segue a condição expressa no método "sacar":

```
24         if(this.saldo >= valor && valor > 0) {  
                ...  
29     }
```

Esta condição pode ser substituída pela assinatura do seguinte contrato:

```
//@ requires this.saldo >= valor && valor > 0;
```

Na tecnologia JML ainda não é possível expressar mensagens personalizadas para o usuário, sendo assim, ao se implementar o contrato em questão, a mensagem "Transação Cancelada: Saldo insuficiente" é descartada.

5.2.4 Exemplo (Parte 4): AspectJML

Evoluindo com o exemplo apresentado na Subseção 5.2.1, abaixo segue a transposição do código AspectJ do aspecto AspectoLogContaCorrente para a sintaxe da linguagem @AspectJ. As diferenças entre os códigos supracitados estão destacados na cor vermelha.

Arquivo "AspectoLogContaCorrente.aj":

```
1 import java.text.DateFormat;
2 import java.text.DecimalFormat;
3 import java.text.SimpleDateFormat;
4 import java.util.Date;
5
6 import org.aspectj.lang.ProceedingJoinPoint;
7 import org.aspectj.lang.annotation.After;
8 import org.aspectj.lang.annotation.Around;
9 import org.aspectj.lang.annotation.Aspect;
10
11 @Aspect
12 public class AspectoLogContaCorrente {
13
14     private DateFormat df = new
15         SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
16     private DecimalFormat formatter = new
17         DecimalFormat("#0.00");
18
19     @After("call(public void ContaCorrente.depositar(..))" +
20         " && target(conta) && args(valor)")
21     public void doLogDeposito(ContaCorrente conta,
22         Double valor) {
23
24         System.out.println("[ " +
25             df.format((new Date()).getTime()) + " ] " +
26             "Depósito - Conta Corrente: " +
27             conta.getNumero() + " - R$ " +
28             formatter.format(valor));
29     }
30
31     @Around("call(public void ContaCorrente.sacar(..)) " +
32         "&& target(conta) && args(valor)")
33     public void doLogSaque(ProceedingJoinPoint joinPoint,
34         ContaCorrente conta, Double valor)
35         throws Throwable {
36
37         if(conta.getSaldo() >= valor && valor > 0) {
38             System.out.println("[ " +
39                 df.format((new Date()).getTime()) +
```

```
40         "]" Saque - Conta Corrente: " +
41         conta.getNumero() + " - Saque: (R$ " +
42         formatter.format(valor).toString() + ")");
43     }
44     joinPoint.proceed();
45
46 }
47
48 }
```

Observando o código do aspecto em estudo (`AspectoLogContaCorrente`), percebe-se uma condição (*if*) no *Advice* `doLogSaque` que pode ser substituída por um contrato DbC.

```
37         if (conta.getSaldo() >= valor && valor > 0) {
43             ...
        }
```

Segue o trecho de código ao ser substituído pela sintaxe AspectJML, que deve ser inserido logo acima da *annotation* `@Around` da chamada do *Advice* `doLogSaque`.

```
//@ requires conta.getSaldo() >= valor && valor > 0;
```

No exemplo acima, o contrato é estabelecido através da pré-condição "requires" exigindo que o saldo, "`conta.getSaldo()`", seja maior ou igual ao valor solicitado para saque e que este valor seja maior que zero. Caso a pré-condição do contrato não seja validada, uma exceção é lançada.

5.2.5 Exemplo (Parte 5): Extensão da AODM

Como pode ser observado no exemplo do modelo AODM expresso na Figura 10, a implementação da condição *Advice* `doLogSaque`, primordial para a execução do *Pointcut* `logSaque`, não pôde ser representada, pois no momento em que este exemplo foi apresentado, essa condição ainda estava implementada em código AspectJ em uma estrutura

condicional (*if*). Mesmo após a transposição do código para AspectJML, como pôde ser visto na Subseção 5.2.4, que transforma essa condição em um contrato DbC, não há como expressar tal regra no padrão de modelagem AODM. Os contratos definidos na Subseção 5.2.3 para a classe `ContaCorrente`, também não podem ser representados no modelo AODM, pois os Diagramas de Classe AODM não dão suporte a contratos DbC em classes, nem em aspectos.

Para atualizar o exemplo da classe `ContaCorrente` e do aspecto `AspectoLogContaCorrente` segue o código atualizado com a implementação dos contratos, cujos contratos DbC estão destacados na cor vermelha:

Arquivo "ContaCorrente.java":

```
1 public class ContaCorrente {
2
3     private String numero;
4     protected Double saldo;
5
6     public String getNumero() {
7         return this.numero;
8     }
9
10    public Double getSaldo() {
11        return this.saldo;
12    }
13
14    public void criaContaCorrente(String numero) {
15        this.numero = numero;
16        this.saldo = 0.0;
17    }
18
19    //@ requires valor > 0;
20    public void depositar(Double valor) {
21        this.saldo = this.saldo + valor;
22    }
23
24    //@ requires this.saldo >= valor && valor > 0;
25    public void sacar(Double valor) {
```

```
26         this.saldo = this.saldo - valor;
27     }
28
29     public void consultaSaldo() {
30         System.out.println("Saldo - Conta Corrente: "
31             + this.numero + " - R$ " + this.saldo);
32     }
33
34 }
```

Arquivo "AspectoLogContaCorrente.aj":

```
1 import java.text.DateFormat;
2 import java.text.DecimalFormat;
3 import java.text.SimpleDateFormat;
4 import java.util.Date;
5
6 import org.aspectj.lang.ProceedingJoinPoint;
7 import org.aspectj.lang.annotation.After;
8 import org.aspectj.lang.annotation.Around;
9 import org.aspectj.lang.annotation.Aspect;
10
11 @Aspect
12 public class AspectoLogContaCorrente {
13
14     private DateFormat df = new
15         SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
16     private DecimalFormat formatter = new
17         DecimalFormat("#0.00");
18
19     @After("call(public void ContaCorrente.depositar(..))" +
20         "&& target(conta) && args(valor)")
21     public void doLogDeposito(ContaCorrente conta,
22         Double valor) {
23
24         System.out.println("[ " +
25             df.format((new Date()).getTime()) + " ] " +
```

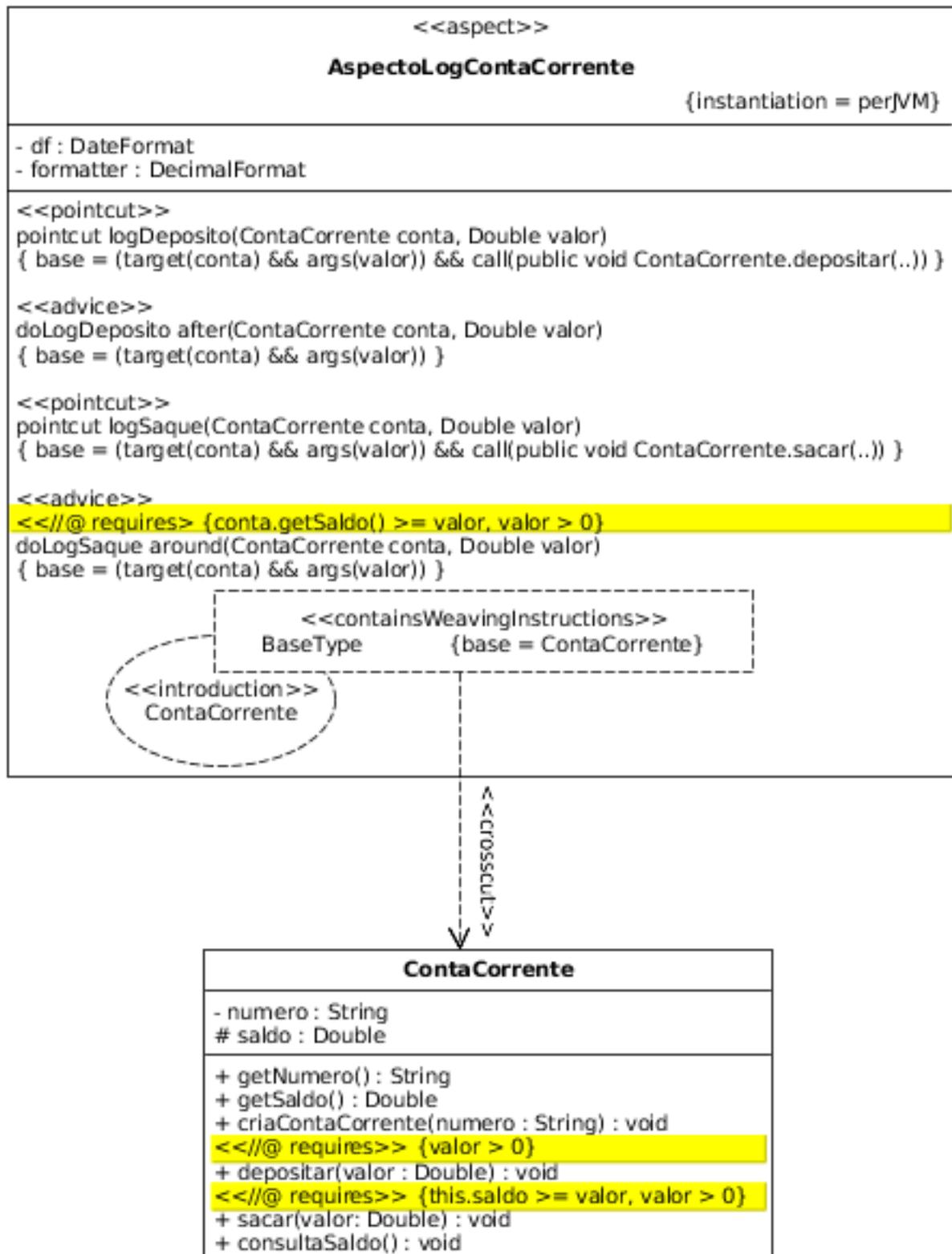
```
26         "Depósito - Conta Corrente: " + conta.getNumero()
27         + - R$ "+ formatter.format(valor));
28
29     }
30
31     //@ requires conta.getSaldo() >= valor && valor > 0;
32     @Around("call(public void ContaCorrente.sacar(..) " +
33         "&& target(conta) && args(valor)")
34     public void doLogSaque(ProceedingJoinPoint joinPoint,
35         ContaCorrente conta, Double valor) throws Throwable {
36
37         System.out.println("[ " + df.format((new
38             Date()).getTime()) + " ] Saque - Conta Corrente: "
39             + conta.getNumero() + " - Saque: (R$ " +
40             formatter.format(valor).toString()+ ")");
41         joinPoint.proceed();
42
43     }
44
45 }
```

Ao implementar o código acima utilizando a apresentação via estereótipos para representar os contratos DbC em classes e aspectos para formar um Diagrama de Classe no padrão estabelecido pela AODM, produziu-se o modelo exposto na Figura 11 onde encontra-se iluminado de amarelo os contratos estabelecidos no código com a sintaxe JML na classe `ContaCorrente` e com a sintaxe `AspectJML` no aspecto `AspectoLogContaCorrente`.

5.3 Avaliação do Artefato

Avaliando a solução descrita na Seção 5.1 e a implementação feita na Seção 5.2, observa-se na Figura 11 um exemplo de um Diagrama de Classe AODM com suporte a contratos DbC. É notório que a implementação não cria nenhum artifício desconhecido pela UML, visto que a cláusula é expressa por estereótipos '`<<//@ ... >>`' e a condição é expressa por propriedade '{ ... }'. Essa característica é um ponto favorável, pois simplifica para as ferramentas CASE (*Computer-Aided Software Engineering*, ou Engenharia de Software

Figura 11 – Modelo AODM Estendido



Fonte: Desenvolvido pelo autor.

Assistida por Computador).² como estas já dão suporte a estereótipos e propriedades, o tra-

² Em função do termo *Computer-Aided Software Engineering* ser muito difundido na literatura técnica brasileira, usaremos o termo em inglês, bem como sua sigla, no decorrer desta dissertação.

balho deve se concentrar em diferenciar os estereótipos comuns '<< ... >>' dos estereótipos DbC '<<//@ ... >>'. Deve-se levar em consideração nesta análise o fato da representação do *pointcut* nativo da AODM ser o único elemento que não pertence nativamente a UML.

Outro ponto positivo é a visibilidade do contrato, pois é expresso logo acima do método, no caso de classes, e logo acima da chamada do *advice*, no caso de aspectos. Esta disposição, separando o estereótipo da cláusula e a propriedade da condição, facilita a leitura do artefato e a sua modelagem também é bastante simples.

Uma questão a se observar é que a escrita do contrato é inspirada na sintaxe do JML, bem como a AODM é inspirada no AspectJ; portanto, corre-se o risco de que em outras implementações de DbC a transposição do código para o artefato não seja tão natural como nas implementações JML e AspectJML. Porém, a simplicidade da escrita não cria empecilhos para transposição de padrões para a escrita de código em outras linguagens.

Um fator crítico é a verbosidade do padrão AODM, que abriga a solução de contratos. Os aspectos em AODM podem ficar grandes ao serem expressos e ao adicionar os contratos, o tamanho dos aspectos podem ficar ainda maiores; logo, as Ferramentas CASE, para simplificar a leitura, poderiam implementar a contração de formas para classes em retângulos com o nome da classe e aspectos em losangos com o nome do aspecto, como em Chavez (2004) e Zakaria, Hosny e Zeid (2002), da periferia do modelo, deixando expandido com leitura completa apenas as classes e aspectos da área em análise.

5.3.1 Parecer Técnico

Como forma de corroborar o processo de avaliação do artefato, seguindo a proposta da *Design Science Research* exposta na Seção 2.3, o artefato desenvolvido nesta dissertação foi submetido a 2 (dois) Pareceres Técnicos de profissionais de Tecnologia da Informação com formação e experiência profissional relevantes na área de desenvolvimento de software, tendo sido avaliado positivamente em ambos os casos, como pode ser observado nos Anexos A e B:

1º Parecer Técnico (Anexo A): Desenvolvido por Soraia Pacheco de Almeida Silva Felício, Doutora em História das Ciências e das Técnicas e Epistemologia pelo HCTE/UFRJ

2º Parecer Técnico (Anexo B): Desenvolvido por Luiz Francisco Dias Pereira, Mestre em Informática pelo NCE/UFRJ e Diretor de Tecnologia da Informação do Colégio Pedro II.

6 CONSIDERAÇÕES FINAIS

A técnica de desenvolvimento de *Design by Contract* vem chamando a atenção dos pesquisadores da área de desenvolvimento de *software* e várias implementações tem sido criadas em linguagens de programação diversas e algumas dessas implementações se estendem à aplicação em AOP. Porém, apesar dos avanços, não era possível modelar os contratos em Diagramas de Classes e assim auxiliar as equipes de análise registrando as premissas dos métodos e *advice* tampouco as condições de saída.

Para se formatar e conduzir a pesquisa, cujo objetivo é estender um artefato com o objetivo específico de expressar contratos DbC em classes e aspectos, optou-se em desenvolver a pesquisa nos padrões da metodologia *Design Science Research*, que vem se apresentado no meio acadêmico, como sendo eficiente no propósito da condução de pesquisas com geração, modificação ou extensão de artefatos.

Após a revisão da literatura e ao avaliar as técnicas de modelagem AOP existentes, optou-se por prosseguir com a pesquisa estendendo o padrão de modelagem AODM, pois nesta técnica, a extensão do Diagrama de Classes permite modelar classes e aspectos em um único artefato e de maneira simples, pois o único artifício alheio a UML para se atingir o seu objetivo é a representação do *pointcut*, todos os demais elementos estão contidos na UML. Para auxiliar na demonstração da técnica de modelagem estendida na pesquisa, optou-se demonstrar o código de Classes com contratos DbC com a sintaxe JML e nos aspectos em AspectJML.

Ao avaliar as técnicas possíveis para representação de *annotation*, optou-se para que a técnica de modelagem desenvolvida nesta pesquisa se baseasse no uso de estereótipos para representar as cláusulas do contrato, juntamente com a chamada do contrato em sintaxe JML `'//@'`, e em propriedades UML para se representar as condições do contrato. Esta sintaxe tem o mesmo comportamento nos métodos das classes e nos *advice* dos aspectos. A simplicidade da representação é um fator preponderante na técnica pois facilita a leitura dos profissionais envolvidos na equipe de desenvolvimento, bem como na escrita no modelo.

O exemplo escrito em Java/AspectJ, criado para se conduzir apresentação do desenvolvimento do artefato foi disposto de forma continuada, para que se demonstrasse a construção do raciocínio demonstrando uma classe de conta-corrente que execute saques e depósitos, com suas regras de funcionamento inseridas dentro do código, tanto da classe quanto do aspecto. Este mesmo exemplo foi modelado em AODM. Numa primeira evolução do exemplo, as condições de funcionamento que estavam enraizadas no código, foram transpostas para JML no caso da classe e em AspectJML no caso do aspecto. Porém,

inicialmente o código do aspecto foi transposto para `@AspectJ`, baseando a sintaxe em *annotation*, para assim respeitar a implementação AspectJML. Quando o código já estava com os contratos implementados nas sintaxes escolhidas para a condução da pesquisa, foi possível por em prática o padrão de modelagem criado para se representar DbC em um novo Diagrama de Classe AODM.

O modelo do exemplo estendido, disposto na Figura 11, apresenta a solução completa, com classes e aspectos com contratos DbC em ambos com a simplicidade e clareza, atendendo os quesitos descritos no objetivo da pesquisa. Com o padrão estabelecido, é possível aplicar em projetos de desenvolvimento de software, independente do porte, levando em consideração que quanto maior o projeto, maior e mais complexo é o modelo resultante, o que pode dificultar a leitura. Em função da AODM ser um padrão verborrágico, os elementos representados nessa técnica ocupam um espaço considerável no artefato, de modo que, é interessante que a sua representação seja feita em meio digital, possibilitando a contração das formas, tanto classes (contração em forma de retângulo), como nos aspectos (contração em forma de losango).

Os resultados estarão dispostos nesta dissertação, disponível na Biblioteca da Universidade Federal de Pernambuco e em formato digital na Biblioteca Digital Brasileira de Teses e Dissertações (BDTD), mantida pelo IBICT (Instituto Brasileiro de Informação em Ciência e Tecnologia).

6.1 Propostas para trabalhos futuros

Após o desenvolvimento do artefato e sua avaliação, para se enriquecer e fundamentar a diagramação de contratos em DbC seguem as propostas para futuros trabalhos de pesquisa:

- Experimentar a solução desenvolvida na pesquisa em projetos desenvolvidos em outras implementações de DbC em Java (outras opções ao JML e AspectJML) e mesmo em outras linguagens de programação;
- Aplicar a solução desenvolvida na pesquisa em ferramentas de modelagem para se facilitar a produção dos modelos AODM, com a extensão para DbC. Inclusive com o recurso de contração das formas para facilitar a análise do artefato despoluindo a periferia do modelo;
- Aplicar a geração de código automático através do modelo AODM com DbC em JML e AspectJML, bem como em outras linguagens de programação;
- Aplicar a engenharia reversa, gerando modelo automaticamente através de um código com DbC;

- Aplicar DbC nos outros padrões de modelos de aspectos, a fim de prover um estudo comparativo dos modelos resultantes, possibilitando assim, a avaliação dos pontos positivos e negativos dos padrões em estudo;

REFERÊNCIAS

- ALDAWUD, O.; ELRAD, T.; BADER, A. A UML Profile for Aspect Oriented Modeling. In: *ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages and Application - OOPSLA - Workshop on Aspect Oriented Programming*. [S.l.: s.n.], 2001. p. 1–6. Citado 2 vezes nas páginas 53 e 59.
- BANNING, J. H. *Ecological Triangulation: An Approach for Qualitative Meta-Synthesis*. [S.l.], 2001. 2 p. Citado 2 vezes nas páginas 50 e 92.
- BARRA, E.; GÉNOVA, G.; LLORENS, J. An Approach to Aspect Modelling with UML 2.0. In: *Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling (AOM)*. [S.l.: s.n.], 2004. Citado 4 vezes nas páginas 52, 53, 54 e 55.
- BARRETO, B. V.; LORENZI, F. Swarm-Forens: Uma Ferramenta para Análise Forense Computacional Baseada na Inteligência Coletiva. p. 1–20, 2011. Citado na página 17.
- BARTER. *Barter - beyond design by contract*. 2002. Disponível em: <<http://barter.sourceforge.net>>. Acesso em: 23/08/2017. Citado na página 21.
- BRERETON, P.; KITCHENHAM, B. A.; BUDGEN, D.; TURNER, M.; KHALIL, M. Lessons From Applying the Systematic Literature Review Process Within the Software Engineering Domain. *Journal of Systems and Software*, v. 80, n. 4, p. 571–583, 2007. ISSN 01641212. Citado 2 vezes nas páginas 37 e 41.
- BUNGE, M. *Emergence and convergence: Qualitative novelty and the unity of knowledge*. Toronto: University of Toronto Press, 2003. 333 p. Citado na página 26.
- CEPA, V.; KLOPPENBURG, S. Representing Explicit Attributes in UML. *Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling (AOM)*, 2005. Citado 3 vezes nas páginas 63, 64 e 87.
- CHAVEZ, C. V. F. *Um Enfoque Baseado em Modelos para o Design Orientado a Aspectos*. 298 p. Tese (Doutorado) — Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2004. Citado 2 vezes nas páginas 53 e 77.
- CLARKE, S. N.; WALKER, R. J. Composition Patterns: An Approach to Designing Reusable Aspects. In: *ACM/IEEE International Conference on Software Engineering - ICSE*. [S.l.: s.n.], 2001. p. 5–14. Citado na página 53.
- COTTENIER, T.; VAN DEN BERG, A.; ELRAD, T. Modeling Aspect-Oriented Compositions. In: *ACM/IEEE International Conference on Model-Driven Engineering Languages and Systems - MODELS*. [S.l.: s.n.], 2006. v. 3844 LNCS, p. 100–109. Citado na página 53.
- DARIO, D.; GUERRERO, S. Design by Contract com JML. *Congresso da Sociedade Brasileira de Computação*, p. 1455–1499, 2005. Citado na página 21.
- DEITEL, H. M.; DEITEL, P. J. *Java Como Programar*. 6. ed. [S.l.]: Pearson Prentice Hall, 2005. 1110 p. Citado na página 57.

- DRESCH, A.; LACERDA, D. P.; ANTUNES JÚNIOR, J. A. V. *Design Science Research: Método de Pesquisa para Avanço da Ciência e Tecnologia*. 1. ed. Porto Alegre, RS - Brasil: Bookman Editora, 2015. 204 p. Citado 8 vezes nas páginas 27, 28, 29, 30, 31, 32, 33 e 52.
- DRESCH, A.; LACERDA, D. P.; MIGUEL, P. A. C. A Distinctive Analysis of Case Study, Action Research and Design Science Research. *Review of Business Management*, p. 1116–1133, 2015. Citado na página 28.
- FELDMAN, Y. A.; BARZILAY, O.; TYSZBEROWICZ, S. Jose: Aspects for Design by Contract. *Workshop on Formal Techniques for Java Programs - FTJP*, 2006. Citado na página 21.
- FOWLER, M.; BECK, K.; BRANT, J.; OPDYKE, W.; ROBERTS, D. *Refactoring: Improving the Design of Existing Code*. [S.l.: s.n.], 1999. 1–337 p. ISSN 14359456. ISBN 9780201485677. Citado na página 22.
- GIBBONS, M.; LIMOGES, C.; NOWOTNY, H.; SCHWARTZMAN, S.; SCOTT, P.; TROW, M. The New Production of Knowledge: The Dynamics of Science and Research in Contemporary Societies. *Contemporary Sociology*, v. 24, p. 751, 1994. Citado na página 27.
- GREENHALGH, T.; PEACOCK, R. Effectiveness and Efficiency of Search Methods in Systematic Reviews of Complex Evidence: Audit of Primary Sources. *Bmj*, v. 331, n. 7524, p. 1064–1065, 2005. Citado na página 46.
- GREGOR, S.; JONES, D. The Anatomy of a Design Science Theory. *Journal of the Association for Information Systems*, v. 8, n. 5, p. 312–335, 2007. Citado na página 27.
- HEVNER, A. R.; MARCH, S. T.; PARK, J.; RAM, S. Design Science in Information Systems Research. *MIS Quarterly*, v. 28, n. 1, p. 75–105, 2004. Citado 3 vezes nas páginas 27, 28 e 30.
- HO, W.-M.; PENNANEAC'H, F.; PLOUZEAU, N. UMLAUT: A Framework for Weaving UML-Based Aspect-Oriented Designs. *International Conference on Technology of Object-Oriented Languages and Systems - TOOLS*, n. Section 2, p. 324–334, 2000. Citado na página 53.
- JOHNSON, R.; HOELLER, J.; DONALD, K.; SAMPALANU, C.; HARROP, R.; RISBERG, T.; DAVISON, D.; KOPYLENKO, D.; POLLACK, M.; TEMPLIER, T.; VERVAET, E.; HALE, B.; COLYER, A.; LEWIS, J.; LEAU, C.; FISHER, M.; BRANNEN, S.; POUTSMA, A.; BEAMS, C.; ABEDRABBO, T.; CLEMENT, A.; SYER, D.; GIERKE, O.; STOYANCHEV, R.; WEBB, P.; WINCH, R.; CLOZEL, B.; NICOLL, S.; DELEUZE, S. *Spring Framework Reference Documentation*. 4.3.10.rel. ed. [S.l.: s.n.], 2016. 884 p. Citado na página 62.
- KHAN, K. S.; TER RIET, G.; GLANVILLE, J.; SOWDEN, A. J.; KLEIJNEN, J. *Undertaking systematic reviews of research on effectiveness: CRD's guidance for carrying out or commissioning reviews*. [S.l.: s.n.], 2001. Citado na página 41.
- KICZALES, G.; HILSDALE, E.; HUGUNIN, J.; KERSTEN, M.; PALM, J.; GRISWOLD, W. G. An Overview of AspectJ. *ECOOP 2001 - Object-Oriented Programming*, v. 2072, n. 4, p. 327–354, 2001. ISSN 03029743. Citado na página 58.
- KICZALES, G.; LAMPING, J.; MENDHEKAR, A.; MAEDA, C.; LOPES, C. V.; LOINGTIER, J.-M.; IRWIN, J. Aspect-Oriented Programming. *Springer-Verlag LNCS*, n. June, 1997. Citado na página 18.

KITCHENHAM, B. A.; CHARTERS, S. Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3. *Engineering*, v. 45, n. 4ve, p. 1051, 2007. ISSN 00010782. Citado 10 vezes nas páginas 34, 35, 36, 37, 42, 48, 86, 89, 92 e 94.

KUHN, T. S. *Estrutura das Revoluções Científicas*. 5. ed. São Paulo: Editora Perspectiva, 1998. 257 p. Citado na página 17.

LEAVENS, G. T. *The Java Modeling Language (JML)*. 1999. Disponível em: <<http://www.jmlspecs.org>>. Acesso em: 31/08/2017. Citado na página 61.

LEAVENS, G. T.; CHEON, Y. Design by Contract with JML. 2006. Citado 2 vezes nas páginas 21 e 62.

LORENZ, D. H.; SKOTINIOTIS, T. Extending Design by Contract for Aspect-Oriented Programming. *ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications*, p. 1–11, 2004. Citado na página 21.

LOSAVIO, F.; MATTEO, A.; MORANTES, P. UML Extensions for Aspect Oriented Software Development. *Journal of Object Technology*, v. 8, n. 5, p. 85–104, 2009. Citado na página 51.

LUFT, L. *Perdas e ganhos*. [S.l.]: Record, 2003. 160 p. Citado na página 7.

MARCH, S. T.; SMITH, G. F. Design and Natural Science Research on Information Technology. *Decision Support Systems*, v. 15, n. 4, p. 251–266, dec 1995. Citado 3 vezes nas páginas 27, 28 e 29.

MEYER, B. Applying "Design Contract". *IEEE Computer*, p. 40–51, 1992. Citado 2 vezes nas páginas 20 e 21.

MEYER, B. *Object-Oriented Software Construction*. 2a. ed. [S.l.]: Prentice-Hall, 2000. Citado na página 61.

MOLDEREZ, T.; JANSSENS, D. Design by Contract for Aspects, by Aspects. *Workshop on Foundations of Aspect-Oriented Languages - FOAL*, p. 9–13, 2012. Citado na página 21.

PEFFERS, K.; TUUNANEN, T.; ROTHENBERGER, M. A.; CHATTERJEE, S. A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, v. 24, n. 3, p. 45–78, 2007. Citado 2 vezes nas páginas 28 e 29.

PIVOTAL SOFTWARE. *Pivotal Software*. 2013. Disponível em: <<https://pivotal.io>>. Acesso em: 31/08/2017. Citado na página 62.

REBÊLO, H. E. M.; LEAVENS, G. T.; BAGHERZADEH, M.; RAJAN, H.; LIMA, R. M. F.; ZIMMERMAN, D. M.; CORNÉLIO, M.; THÜM, T. *AspectJML crosscutting contract specifications for better modularity*. 2014. Disponível em: <<http://www.cin.ufpe.br/~hemr/aspectjml>>. Acesso em: 31/08/2017. Citado na página 62.

REBÊLO, H. E. M.; LEAVENS, G. T.; BAGHERZADEH, M.; RAJAN, H.; LIMA, R. M. F.; ZIMMERMAN, D. M.; CORNÉLIO, M.; THÜM, T. AspectJML: Modular Specification and Runtime Checking for Crosscutting Contracts. In: *International Conference on Modularity*. [S.l.: s.n.], 2014. v. 13, p. 157–168. Citado na página 62.

- REBÊLO, H. E. M.; LEAVENS, G. T.; LIMA, R. M. F.; BORBA, P.; RIBEIRO, M. Modular Aspect-Oriented Design Rule Enforcement with XPIDRs. *Workshop on Foundations of Aspect-Oriented Languages - FOAL*, p. 13, 2013. Citado na página 21.
- ROMME, G. L. Making a Difference: Organization as Design. *Organization Science*, v. 14, n. 5, p. 558–573, 2003. Citado na página 27.
- ROSENHAINER, L. Identifying Crosscutting Concerns in Requirements Specifications. *Early Aspects Workshop 2004: Aspect-Oriented Requirements Engineering and Architecture Design, OOPSLA 2004*, p. 49–58, 2004. Citado na página 17.
- RUMBAUGH, J.; JACOBSON, I.; BOOCH, G. *The Unified Modeling Language Reference Manual*. 2nd editio. ed. [S.l.]: Addison Wesley Longman, Inc, 1999. 550 p. Citado 2 vezes nas páginas 20 e 37.
- SANDELOWSKI, M.; VOILS, C. I.; LEEMAN, J.; CRANDELL, J. L. Mapping the Mixed Methods–Mixed Research Synthesis Terrain. *Journal of Mixed Methods Research*, v. 6, n. 4, p. 317–331, 2012. Citado 2 vezes nas páginas 38 e 90.
- SCHREIBER, G.; AKKERMANS, H.; ANJEWIERDEN, A.; HOOG, R. de; SHADBOLT, N.; VELDE, W. V. de; WIELINGA, B. *Knowledge Engineering and Management: The CommonKADS Methodology*. [S.l.]: Massachusetts Institute of Technolog, 2000. 456 p. ISBN 0262193000. Citado na página 26.
- SIMON, H. A. *The Sciences of the Artificial*. 3. ed. [S.l.]: MIT Press, 1996. 241 p. Citado 2 vezes nas páginas 26 e 27.
- SKOTINIOTIS, T.; LORENZ, D. H. Cona: Aspects for Contracts and Contracts for Aspects. *ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications*, p. 196–197, 2004. Citado na página 21.
- SKOTINIOTIS, T.; LORENZ, D. H. *Conaj: Generating Contracts as Aspects*. [S.l.], 2004. 20 p. Citado na página 21.
- SORDI, J. O. D.; MEIRELES, M.; SANCHES, C. Design Science: Uma Abordagem Inexplorada por Pesquisadores Brasileiros em Gestão de Sistemas de Informação. *XXXIV Encontro da ANPAD*, p. 1–15, 2010. Citado na página 30.
- SPRING FRAMEWORK. *Spring Framework*. 2002. Disponível em: <<https://projects.spring.io/spring-framework/>>. Acesso em: 31/08/2017. Citado na página 62.
- STEIN, D.; HANENBERG, S.; UNLAND, R. A UML-Based Aspect-Oriented Design Notation for AspectJ. In: *Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling With UML (AOM)*. [S.l.: s.n.], 2002. p. 106. Citado na página 60.
- STEIN, D.; HANENBERG, S.; UNLAND, R. Designing Aspect-Oriented Crosscutting in UML. In: *Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling With UML (AOM)*. [S.l.: s.n.], 2002. Citado 2 vezes nas páginas 53 e 59.
- TESSER, G. J. Principais Linhas Epistemológicas Contemporâneas. *Educ. rev.*, Curitiba, v. 10, p. 91–98, 1994. Citado na página 28.

- VAISHNAVI, V.; KUECHLER, B. Design Science Research in Information Systems. *Ais*, p. 45, 2004. Citado 2 vezes nas páginas 28 e 29.
- VAN AKEN, J. E. Management Research Based on the Paradigm of the Design Sciences: The Quest for Field-Tested and Grounded Technological Rules. *Journal of Management Studies*, v. 41, n. 2, p. 219–246, 2004. Citado 2 vezes nas páginas 27 e 28.
- VAN AKEN, J. E. Management Research as a Design Science: Articulating the Research Products of Mode 2 Knowledge Production in Management. *British Journal of Management*, v. 16, n. 1, p. 19–36, mar 2005. Citado 2 vezes nas páginas 27 e 34.
- VEIT, D. R.; LACERDA, D. P.; CAMARGO, L. F. R.; KIPPER, L. M.; DRESCH, A. Towards Mode 2 knowledge production: analysis and proposal of a framework for research in business processes. *Business Process Management Journal*, v. 23, n. 2, p. 1–22, 2017. Citado na página 28.
- WALLS, J. G.; WIDMEYER, G. R.; SAWY, O. A. E. Building an Information System Design Theory for Vigilant EIS. *Information Systems Research*, v. 3, n. 1, p. 36–59, 1992. ISSN 10477047. Citado na página 27.
- WAMPLER, D. Contract4J for Design by Contract in Java: Design Pattern-Like Protocols and Aspect Interfaces. *AOSD Workshop on Aspects, Components, and Patterns for Infrastructure Software*, p. 27–30, 2006. Citado na página 21.
- WANG, Y.; SINGH, S.; HOSKING, J.; GRUNDY, J. An Aspect-Oriented UML Tool for Software Development with Early Aspects. In: *International Workshop on Early Aspects at ICSE*. [S.l.: s.n.], 2006. p. 51. Citado na página 53.
- WIERINGA, R. *Design Science Methodology for Information Systems and Software Engineering*. Berlin: [s.n.], 2014. 493 p. Citado na página 26.
- ZAKARIA, A. A.; HOSNY, H.; ZEID, A. A UML Extension for Modeling Aspect-Oriented Systems. In: *Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling With UML (AOM)*. [S.l.: s.n.], 2002. Citado 5 vezes nas páginas 52, 53, 54, 55 e 77.
- ZHAO, J.; RINARD, M. Pipa: A Behavioral Interface Specification Language for AspectJ. *International Conference on Fundamental Approaches to Software Engineering (FASE)*, p. 1–15, 2003. Citado na página 21.

APÊNDICE A – PROTOCOLO DA PESQUISA CONFORME DESIGN SCIENCE RESEARCH

A.1 Identificação do Problema

A técnica de *Design by Contract* é eficaz para a delimitação da regra para uso do método, porém não há como modelá-lo no padrão da UML, devido ao fato deste padrão de modelagem ser restrito à OO e à DbC ser uma extensão desta metodologia.

O problema de não se ter como modelar DbC se apresenta ainda maior ao se estender para Aspectos, como em AspectJML.

A.2 Conscientização do Problema

Em função da POA não estar contida na UML, alguns estudos foram feitos nesse sentido e algumas extensões para a UML foram elaboradas ao longo do tempo. Ainda hoje não há um consenso sobre uma técnica que tivesse virado um padrão de mercado. E como é percebido na "Identificação do Problema", apesar de algumas técnicas para modelar a POA, não é possível a modelagem dos contratos do DbC.

Os benefícios da criação de uma técnica nesses moldes seria muito útil para equipes de desenvolvimento que implementam Aspectos em seu código, independente da linguagem de programação ou do *framework* POA adotado, possibilitando uma maior clareza na prática de desenvolvimento paralelo, com a demonstração das regras das interfaces dos aspectos de forma visual, facilitando a demonstração do projeto para a equipe de desenvolvimento, tornando as decisões mais ágeis e evitando retrabalhos. Com a criação da técnica de modelar DbC passa a ser possível o desenvolvimento de módulos que apliquem essa técnica nos sistemas de desenvolvimento UML de mercado e, enfim, possibilitem Engenharia Reversa para a geração de modelos de sistemas legados (OO / POA / DbC).

A.3 Revisão Sistemática da Literatura

O Protocolo da Revisão Sistemática da Literatura dessa dissertação foi desenvolvido com base em Kitchenham e Charters (2007) e está descrita no Apêndice B

A.4 Identificação dos Artefatos e Configuração das Classes de Problemas

O artefato identificado é uma extensão do Diagrama de Classe que consiga expressar os contratos estabelecidos pela DbC em Classes e Aspectos.

A classe de problema é a falta de condições de se expressar os referidos contratos nos padrões de modelo de Diagrama de Classe existentes, bem como em suas extensões.

A.5 Proposição de Artefatos para Resolver o Problema Específico

Como proposição de Artefato, optou-se por uma extensão do Diagrama de Classes do padrão estabelecido pela *Aspect Oriented - Unified Model Language* (AODM), adicionando recursos para se expressar contratos

A.6 Projeto do Artefato Selecionado

A solução encontrada para se adicionar recursos para se expressar contratos DbC em Diagrama de Classes da AODM baseia-se no estudo de Cepa e Kloppenburg (2005) que avaliou as formas possíveis de se expressar *annotations* em Diagrama de Classes. Adaptando para contratos e extendendo para aspectos, optou-se pelo formato de cláusulas de contratos em estereótipos <<//@ ... >> seguido da notação de propriedade UML com a condição do contrato { ... }, conforme descrito abaixo:

```
<<//@ cláusula>> {condição}
```

A.7 Desenvolvimento do Artefato

Na Figura 11 está expresso um exemplo da extensão do Diagrama de Classes AODM que dê suporte a DbC. Nesta figura está expresso um contrato DbC no *advice* doLogSaque do aspecto AspectoLogContaCorrente, outros 2 (dois) contratos nos métodos depositar e sacar da classe ContaCorrente. Como forma de destacar os exemplos, os contratos estão iluminados de amarelo tanto na classe quanto no aspecto.

A.8 Avaliação do Artefato

O artefato sofreu uma avaliação própria que está explicitado na Seção A.9 e uma avaliação feita por 3 (três) Pareceres Técnicos por profissionais com formação e experiência profissional na área de desenvolvimento de software, dispostos nos Anexos A e B.

A.9 Explicitação das Aprendizagens

O artefato foi avaliado com os seguintes pontos positivos:

- Simplicidade de implementação, devido ao fato de que para se criar a solução nenhum artifício desconhecido pela UML foi incorporado;

- Boa visibilidade em função da disposição acima de métodos, no caso das classes e das chamadas dos *advice* no caso dos aspectos;
- Outro fator de visibilidade foi em função da clareza da disposição das cláusulas em estereótipos e condições em propriedades UML;

Fatores críticos:

- A sintaxe inspirada na JML e na AspectJML trás uma simplicidade para o entendimento; porém, a transposição dos contratos escritos em outras linguagens pode não ser tão natural;
- Em função da solução ser verborrágica estendendo a técnica AODM que também é verborrágica, a descrição dos aspectos, juntamente com as classes no mesmo diagrama pode-se gerar um Diagrama de Classes grande. Porém há a possibilidade de ao se manipular digitalmente o artefato, trabalhar com contração de formas.

A.10 Conclusões

A conclusão da pesquisa é positiva, devido aos objetivos pré-estabelecidos na dissertação terem sido atingidos. Foram propostos caminhos para trabalhos futuros.

A.11 Generalização para uma Classe de Problemas

O padrão de modelo de DbC desenvolvido na pesquisa permite a sua aplicação em projetos de desenvolvimento de software independente do seu porte.

A.12 Comunicação dos Resultados

Esta dissertação permanecerá disponível na Biblioteca Institucional da Universidade Federal de Pernambuco e em formato digital na BDTD.

APÊNDICE B – PROTOCOLO DA RSL

B.1 Planejamento da Revisão

O Planejamento da Revisão é a primeira etapa da Revisão Sistemática da Literatura. Nesta etapa planeja-se e prepara-se toda a estratégia da revisão.

B.1.1 Identificação da Necessidade da Revisão

Com a UML é possível modelar a OO, mas com o padrão mantida pela OMG só é possível modelar POA através de algumas técnicas independentes publicadas ao longo do tempo. Porém, nenhuma dessas propostas abrangem a técnica *Design by Contract*, trazendo prejuízos a equipes de desenvolvimento de software que necessitam representar graficamente DbC.

B.1.2 Comissionamento da Revisão

Não haverá comissionamento para a execução da Revisão Sistemática da Literatura nesta dissertação, visto que o próprio autor executará essa atividade.

B.1.3 Questão de Pesquisa

"Como definir uma técnica de modelagem de Diagrama de Classe, com base na UML, e em uma extensão para POA, que possibilite a representação gráfica da técnica Design by Contract?"

B.1.4 Protocolo da Revisão

O Protocolo da Revisão Sistemática da Literatura desta pesquisa foi desenvolvido com base em Kitchenham e Charters (2007) e está descrita neste Apêndice.

B.1.5 Avaliação do protocolo de revisão

A realização da avaliação do protocolo foi feita pelo orientador desta dissertação, o Prof. Dr. Henrique Emanuel Mostaert Rebêlo durante o período de orientações na escrita da dissertação.

B.2 Condução da Revisão

A Condução da Revisão é a segunda etapa da Revisão Sistemática da Literatura. Nesta etapa executa-se a revisão propriamente dita, com base nos levantamentos e conclusões resultantes da Etapa de Planejamento da Revisão.

B.2.1 Identificação da Pesquisa

A Estratégia da Revisão Sistemática da Literatura desta pesquisa deverá obedecer os seguintes critérios:

Horizonte Temporal: Estudos aplicados nos últimos 20 (vinte) anos, ou seja a partir de 1997 (mil novecentos e noventa e sete). A amplitude do "Horizonte Temporal" deu-se devido ao fato de haver estudos muito importante na área no final dos anos 1990 (mil novecentos e noventa).

Idiomas: Estudos publicados em português ou inglês.

Estratégia de Revisão: Como a questão da pesquisa visa tratar de um problema em aberto, a melhor estratégia é a adoção da "Revisão Configurativa" que, segundo Sandelowski *et al.* (2012), tendem a ser respondidas com dados qualitativos, extraídos de estudos primários heterogêneos.

Termos de Busca: A elaboração do termo padrão de busca para a pesquisa levou em consideração os idiomas selecionados pela "Identificação da Pesquisa"; portanto, deve ser produzida uma *string* de termos de busca para cada idioma, porém foi elaborada uma única *string* que atende aos idiomas selecionados. O termo de busca está baseado em siglas, que são comuns a outras áreas, como saúde, geografia, educação, entre outras; logo, optou-se por um bloco de restrição (NOT) dentro do termo de busca, a fim de melhorar a qualidade do resultado. Este termo padrão deve ser adaptado para execução em cada sistema de busca.

```
((model* OR UML OR Motorola) NEAR/3 (Aspect* OR AOP OR POA)) OR "Design by Contract") NOT (mathemat* OR hospit* OR biolog* OR health* OR medic* OR *chemic* OR marketing OR food OR restaurant OR drink OR Meteorolog* OR Psycholog* OR electr* OR merchandising OR social OR temperature OR physics OR mechatronic* OR philosoph* OR alcohol OR mechanic* OR physiolog* OR Intertextuality OR acoustic OR pollution OR pedagog* OR "verb phrase" OR gis OR telecommunication OR cellular OR adolescen* OR youth* OR clinic* OR disease OR ecolog* OR geolog* OR financ* OR economic* OR *ergonomic* OR ecosystem* OR crime OR morpholog* OR anatom* OR school* OR war)
```

Índice de Busca: As partes das publicações que serão avaliadas pela "Identificação da Pesquisa" são *Abstract*, *Resumo* e *Keywords*.

Fontes de Busca: Foram utilizados Bases de Dados e site de busca da internet.

Biblioteca Digital Brasileira de Teses e Dissertações: <http://bdtd.ibict.br/vufind/>

EBSCOhost <https://www.ebscohost.com>

Google Scholar: <https://scholar.google.com.br>

Scopus <http://www.scopus.com>

SciELO <http://www.scielo.org>

B.2.2 Seleção da Pesquisa Primária

A Seleção da Pesquisa Primária levará em consideração critérios de inclusão e exclusão que devem ser aplicados aos "índices de busca" selecionados durante a "identificação da pesquisa":

Critérios de Inclusão: São considerados critérios de inclusão as seguintes características:

- Trabalho sobre Modelagem POA que estendam a UML;
- Trabalho sobre *Design by Contract*;
- Estudo de Caso sobre *Design by Contract*;
- Revisão Sistemática da Literatura sobre POA.

Critérios de Exclusão: São considerados critérios de exclusão as seguintes características:

- Trabalho sobre Modelagem POA independente da UML;
- Trabalho sobre outras extensões da UML;
- Trabalho sobre outras extensões da POA;
- Revisão Sistemática da Literatura sobre extensões de POA;
- Estudo de Caso sobre UML.

B.2.3 Avaliação da Qualidade do Estudo

A fim de classificar a relevância dos estudos levantados na Seleção da Pesquisa Primária, com o objetivo de direcionar a atenção do pesquisador durante o trabalho, deve-se avaliar o índice adotado na "Identificação da Pesquisa", e categorizar os trabalhos selecionados com os seguintes conceitos:

- Alta Relevância;
- Média Relevância;
- Baixa Relevância.

B.2.4 Extração e Monitoramento dos Dados

Para se catalogar os estudos selecionados Kitchenham e Charters (2007) orienta a criação de um formulário; porém, para se automatizar o processo, foi desenvolvido um dicionário de dados para se extrair os seus meta-dados.

Quadro 11 – Dicionário de Dados dos Estudos

Atributo	Domínio	Tamanho	Descrição
ID	Texto	50	Identificador do trabalho, composto por "sobrenome do autor" + "ano de publicação" + "letra caso haja mais de uma publicação deste autor no mesmo ano"
Título	Texto	255	Título do Trabalho Científico.
Tipo de Publicação	Texto	50	Exemplo: Artigo, Livro, Tese, etc..
Autores	Array[Texto]	50	Lista dos autores do trabalho
Palavras-Chave	Array[Texto]	50	Palavras descritas como "Palavras-Chave" ou "Keywords" do trabalho
Ano de Publicação	Inteiro	9999	Ano de Publicação do trabalho
Periódico	Texto	50	Nome do periódico que foi publicado o trabalho
Tag	Texto	50	Campo reservado para classificação textual do pesquisador. Exemplo: <i>Design by Contract</i> , Modelagem POA, etc...
Grau de Relevância	Texto	5	Alta, Baixa ou Média

Fonte: Desenvolvido pelo autor.

B.2.5 Síntese dos Dados

A Síntese dos Dados foi elaborada de acordo com a técnica de "triangulação ecológica", desenvolvida por Banning (2001) que, ao se avaliar as técnicas de AOM levantadas pela RSL, optou-se pelo uso da técnica AODM e ao se avaliar as técnicas de DbC sobre Aspectos, optou-se pelo uso da sintaxe do AspectJML.

B.3 Documentação da Revisão

A Documentação da Revisão Sistemática da Literatura se dará pela publicação da presente dissertação na biblioteca institucional da Universidade Federal de Pernambuco.

B.3.1 Especificação dos Mecanismos de Disseminação

Em princípio, o mecanismo de disseminação dos resultados desta Revisão Sistemática da Literatura será por meio da própria dissertação que motivou o trabalho de revisão

ficando disponível na biblioteca da UFPE, e de seus repositórios digitais.

B.3.2 Formatação do Relatório Principal

A formatação da publicação respeita os padrões impostos pela ABNT.

B.3.3 Avaliação do Relatório

O modelo de avaliação da publicação é por meio de uma banca examinadora escolhida pela UFPE.

APÊNDICE C – LISTAGEM DAS PUBLICAÇÕES SELECIONADAS PELA RSL

O Quadro 12, com a listagem das publicações selecionadas para esta dissertação, foi elaborado de acordo com a Revisão Sistemática da Literatura, conforme modelo desenvolvido por Kitchenham e Charters (2007) e segue a formatação disposta no Dicionário de Dados elaborado no Protocolo da RSL conforme descrito no Quadro 11.

Quadro 12 – Listagem das Publicações Selecionadas pela RSL

Atributo	Publicação
ID	Ciraci2010
Título	<i>A Graph-Based Aspect Interference Detection Approach for UML-Based Aspect-Oriented Models</i>
Tipo de Publicação	Livro
Autores	CIRACI, Selim; HAVINGA, Wilke; AKSIT, Mehmet; BOCKISCH, Christoph; VAN DEN BROEK, Pim
Palavras-Chave	-
Ano de Publicação	2010
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD)</i>
Tag	AOM
Grau de Relevância	Médio
ID	Evermann2007
Título	<i>A Meta-Level Specification and Profile for AspectJ in UML</i>
Tipo de Publicação	Artigo de Periódico
Autores	EVERMANN, Joerg
Palavras-Chave	-
Ano de Publicação	2007
Periódico	<i>Journal of Object Technology</i>
Tag	Extensão UML: AODM
Grau de Relevância	Alto
ID	Chavez2002
Título	<i>A Metamodel for Aspect-Oriented Modeling</i>
Tipo de Publicação	Artigo de Conferência
Autores	CHAVEZ, Christina Von Flach; LUCENA, Carlos
Palavras-Chave	-
Ano de Publicação	2002
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling With UML (AOM)</i>

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
Tag	Extensão UML: aSideML
Grau de Relevância	Alto
ID	Evermann2011
Título	<i>A Platform-Independent UML Profile for Aspect-Oriented Development</i>
Tipo de Publicação	Artigo de Conferência
Autores	EVERMANN, Joerg; FIECH, Adrian; ALAM, Farhana Eva
Palavras-Chave	<i>Aspect-Oriented Modeling; Aspect-Oriented System Development; UML Profile</i>
Ano de Publicação	2011
Periódico	<i>International C* Conference on Computer Science & Software Engineering - C3S2E</i>
Tag	Extensão UML: AODM
Grau de Relevância	Médio
ID	Li2004
Título	<i>A Process to Combine AOM and AOP: A Proposal Based on a Case Study</i>
Tipo de Publicação	Artigo de Conferência
Autores	LI, Jingyue; HOUMB, Siv Hilde; KVALE, Axel Anders
Palavras-Chave	<i>Aspect-Oriented Development (AOD); Aspect-Oriented Programming (AOP); Aspect-oriented Modeling (AOM); Model-Driven Development (MDD)</i>
Ano de Publicação	2004
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling With UML (AOM)</i>
Tag	AOM
Grau de Relevância	Baixo
ID	Schauerhuber2006
Título	<i>A Survey on Aspect-Oriented Modeling Approaches</i>
Tipo de Publicação	Relatório
Autores	SCHAUERHUBER, Andrea; SCHWINGER, Wieland; KAPSAMMER, Elisabeth; RETSCHITZEGGER, Werner; WIMMER, Manuel; KAPPEL, Gerti
Palavras-Chave	-
Ano de Publicação	2006
Periódico	<i>Vienna University of Technology</i>

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
Tag	Extensão UML: AODM
Grau de Relevância	Alto
ID	Wimmer2011
Título	<i>A Survey on UML-Based Aspect-Oriented Design Modeling</i>
Tipo de Publicação	Artigo de Periódico
Autores	WIMMER, Manuel; SCHAUERHUBER, Andrea; KAPPEL, Gerti; RETSCHITZEGGER, Werner; SCHWINGER, Wieland; KAPSAMMER, Elisabeth
Palavras-Chave	<i>Aspect-Orientation; Model-Driven Development</i>
Ano de Publicação	2011
Periódico	<i>Journal ACM Computing Surveys</i>
Tag	Extensão UML: AODM
Grau de Relevância	Alto
ID	Ho2002
Título	<i>A Toolkit for Weaving Aspect Oriented Designs</i>
Tipo de Publicação	Artigo de Conferência
Autores	HO, Wai-Ming; JÉZÉQUEL, Jean-Marc; PENNANEAC'H, François; PLOUZEAU, Noël
Palavras-Chave	-
Ano de Publicação	2002
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling With UML (AOM)</i>
Tag	Extensão UML: UMLAUT
Grau de Relevância	Médio
ID	Vachon2004
Título	<i>A UML Aspect-Oriented Modeling Approach for Model-Driven Software Development</i>
Tipo de Publicação	Artigo de Conferência
Autores	VACHON, Julie; MOSTEFAOUI, Farida
Palavras-Chave	<i>Aspect-Oriented Modeling; Development Process; MDA; Model Transformation; Non Funcional Requirements; UML profile</i>
Ano de Publicação	2004
Periódico	<i>International Conference on UML Modeling Languages and Applications</i>
Tag	Extensão UML: AODM
Grau de Relevância	Alto

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
ID	Zakaria2002
Título	<i>A UML Extension for Modeling Aspect-Oriented Systems</i>
Tipo de Publicação	Artigo de Conferência
Autores	ZAKARIA, Aida Atef; HOSNY, Hoda; ZEID, Amir
Palavras-Chave	-
Ano de Publicação	2002
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling With UML (AOM)</i>
Tag	Extensão UML: Outras Técnicas
Grau de Relevância	Alto
ID	Pawlak2002
Título	<i>A UML Notation for Aspect-Oriented Software Design</i>
Tipo de Publicação	Artigo de Conferência
Autores	PAWLAK, Renaud; DUCHIEN, Laurence; GERARD, Florin; LEGOND-AUBRY, Fabrice; SEINTURIER, Lionel; MARTELLI, Laurent
Palavras-Chave	-
Ano de Publicação	2002
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling With UML (AOM)</i>
Tag	Extensão UML: AODM
Grau de Relevância	Médio
ID	Aldawud2001
Título	<i>A UML Profile for Aspect Oriented Modeling</i>
Tipo de Publicação	Artigo de Conferência
Autores	ALDAWUD, Omar; ELRAD, Tzilla; BABER, Atef
Palavras-Chave	-
Ano de Publicação	2001
Periódico	<i>ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages and Application - OOPSLA - Workshop on Aspect Oriented Programming</i>
Tag	Extensão UML: AODM
Grau de Relevância	Médio
ID	Stein2002
Título	<i>A UML-Based Aspect-Oriented Design Notation for AspectJ</i>
Tipo de Publicação	Artigo de Conferência

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
Autores	Stein, Dominik; Hanenberg, Stefan; Unland, Rainer
Palavras-Chave	-
Ano de Publicação	2002
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling With UML (AOM)</i>
Tag	Extensão UML: AODM
Grau de Relevância	Baixo
ID	Barra2004a
Título	<i>An Approach to Aspect Modelling with UML 2.0</i>
Tipo de Publicação	Artigo de Conferência
Autores	BARRA, Eduardo; GÉNOVA, Gonzalo; LLORENS, Juan
Palavras-Chave	<i>Advance; Aspect-Oriented Programming; Aspect-Oriented Software Development; Aspects Conceptual Model; Aspects Modelling; Separation of Concerns; UML 2.0</i>
Ano de Publicação	2004
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling (AOM)</i>
Tag	Extensão UML: Outras Técnicas
Grau de Relevância	Alto
ID	Simmonds2005
Título	<i>An Aspect Oriented Model Driven Framework</i>
Tipo de Publicação	Artigo de Conferência
Autores	SIMMONDS, Devon; SOLBERG, Arnor; REDDY, Raghu; FRANCE, Robert; GHOSH, Sudipto
Palavras-Chave	<i>Aspect-Oriented Software Development; QVT; UML; Distributed Applications; Middleware; Model Driven Development; Separation of Concerns; Transactions</i>
Ano de Publicação	2005
Periódico	<i>IEEE International Enterprise Distributed Object Computing Workshop - EDOC</i>
Tag	Extensão UML: UMLAUT
Grau de Relevância	Alto
ID	Wang2006
Título	<i>An Aspect-Oriented UML Tool for Software Development with Early Aspects</i>
Tipo de Publicação	Artigo de Conferência

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
Autores	WANG, Yang; SINGH, Santokh; HOSKING, John; GRUNDY, John
Palavras-Chave	AO-UML; AOCE; <i>Early Aspects; Meta-Modeling Tool</i>
Ano de Publicação	2006
Periódico	<i>International Workshop on Early Aspects at ICSE</i>
Tag	Extensão UML: Outras Técnicas
Grau de Relevância	Alto
ID	Magableh2012a
Título	<i>An Aspectual UML Modelling Tool</i>
Tipo de Publicação	Artigo de Periódico
Autores	MAGABLEH, Aws A.; SHUKUR, Zarina; ALI, Noorazeen Mohd
Palavras-Chave	AOM; <i>Aspect Modelling; Aspect Representations; Aspect-Oriented UML; AspectJ; Aspects; Aspectual UML Diagrams; Crosscutting Concerns</i>
Ano de Publicação	2012
Periódico	<i>Journal of Applied Sciences</i>
Tag	RSL
Grau de Relevância	Médio
ID	Masconi2008
Título	<i>Applying and Evaluating AOM for Platform Independent Behavioral UML Models</i>
Tipo de Publicação	Artigo de Conferência
Autores	MOSCONI, Marco; CHARFI, Anis; SVACINA, Jaroslav; WLOKA, Jan
Palavras-Chave	MDA; <i>UML 2 Profiles; Aspect-Oriented Modeling; Model Weaving</i>
Ano de Publicação	2008
Periódico	Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling (AOM)
Tag	Extensão UML: WEAVR
Grau de Relevância	Alto
ID	Zhang2007
Título	<i>Aspect Composition in the Motorola Aspect-Oriented Modeling Weaver</i>
Tipo de Publicação	Artigo de Periódico
Autores	ZHANG, Jing; COTTENIER, Thomas; VAN DEN BERG, Aswin; GRAY, Jeff
Palavras-Chave	-

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
Ano de Publicação	2007
Periódico	<i>Journal of Object Technology</i>
Tag	Extensão UML: WEAVR
Grau de Relevância	Alto
ID	Nouh2010
Título	<i>Aspect Weaver: A Model Transformation Approach for UML Models</i>
Tipo de Publicação	Artigo de Conferência
Autores	NOUH, M.; ZIARATI, R.; MOUHEB, Djedjiga; ALHADIDI, D.; DEBBABI, Mourad; WANG, Lingyu; POURZANDI, Makan
Palavras-Chave	-
Ano de Publicação	2010
Periódico	Conference of the Center for Advanced Studies on Collaborative Research - CASCON
Tag	AOM
Grau de Relevância	Médio
ID	Rodriguez2014
Título	<i>Aspect-Enhanced Goal-Driven Sequence Diagram</i>
Tipo de Publicação	Artigo de Periódico
Autores	LEE, Jonathan; WU, Chia-Ling; LEE, Wen-Tin; HSU, Kuo-Hsun
Palavras-Chave	-
Ano de Publicação	2014
Periódico	<i>International Journal of Intelligent Systems</i>
Tag	Extensão UML: AODM
Grau de Relevância	Baixo
ID	Kienzle2010
Título	<i>Aspect-Oriented Design with Reusable Aspect Models</i>
Tipo de Publicação	Trecho de Livro
Autores	KIENZLE, Jörg; AL ABED, Wisam; FLEUREY, Franck; JÉZÉQUEL, Jean-Marc; KLEIN, Jacques
Palavras-Chave	-
Ano de Publicação	2010
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD)</i>
Tag	AOM
Grau de Relevância	Médio
ID	Ancha2015

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
Título	<i>Aspect-Oriented Formal Modeling: (AspectZ + Object-Z) = OO-AspectZ</i>
Tipo de Publicação	Artigo de Periódico
Autores	SILVA, Cristian Vidal; VILLARROEL, Rodolfo; SIMÓN, Rodolfo Schmal; SAENS, Rodrigo; TIGERO, Tamara; DEL RIO, Carolina
Palavras-Chave	AspectZ; Object-Z; Z; <i>Aspects</i> ; <i>Crosscutting Concerns</i>
Ano de Publicação	2015
Periódico	<i>Computing and Informatics</i>
Tag	Geração de Código
Grau de Relevância	Baixo
ID	Mehmood2013
Título	<i>Aspect-Oriented Model-Driven Code Generation: A Systematic Mapping Study</i>
Tipo de Publicação	Artigo de Periódico
Autores	MEHMOOD, Abid; JAWAWI, Dayang N. A.
Palavras-Chave	<i>Aspect-Oriented Software Development</i> ; <i>Code Generation</i> ; <i>Model-Driven Engineering</i> ; <i>Systematic Map</i>
Ano de Publicação	2013
Periódico	<i>Information and Software Technology</i>
Tag	Geração de Código
Grau de Relevância	Baixo
ID	Groher2009
Título	<i>Aspect-Oriented Model-Driven Software Product Line Engineering</i>
Tipo de Publicação	Trecho de Livro
Autores	GROHER, Iris; VOELTER, Markus
Palavras-Chave	<i>Software Product Line Development</i> ; <i>Aspect-Oriented Software Development</i> ; <i>Model-Driven Software Development</i>
Ano de Publicação	2009
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD)</i>
Tag	AOM
Grau de Relevância	Médio
ID	Debbabi2011
Título	<i>Aspect-Oriented Modeling for Representing and Integrating Security Concerns in UML</i>
Tipo de Publicação	Artigo de Periódico

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
Autores	DEBBABI, Mourad; TALHI, Chamseddine; NOUH, M.; LIMA, Vitor; WANG, Lingyu; POURZANDI, Makan
Palavras-Chave	<i>Access Control; Aspect-Oriented Modeling; Security Enforcement; Security Requirements; UML</i>
Ano de Publicação	2011
Periódico	<i>ACIS International Journal of Computer & Information Science</i>
Tag	Extensão UML: AODM
Grau de Relevância	Baixo
ID	Elrad2002
Título	<i>Aspect-Oriented Modeling: Bridging the Gap between Implementation and Design</i>
Tipo de Publicação	Artigo de Periódico
Autores	ELRAD, Tzilla; ALDAWUD, Omar; BADER, Atef
Palavras-Chave	-
Ano de Publicação	2002
Periódico	<i>Lecture Notes in Computer Science</i>
Tag	Extensão UML: AODM
Grau de Relevância	Alto
ID	Richter2003
Título	<i>Aspect-Oriented Monitoring of UML and OCL Constraints</i>
Tipo de Publicação	Artigo de Conferência
Autores	RICHTERS, Mark; GOGOLLA, Martin
Palavras-Chave	-
Ano de Publicação	2003
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling With UML (AOM)</i>
Tag	Extensão UML: Outras Técnicas
Grau de Relevância	Médio
ID	Araujo2002
Título	<i>Aspect-Oriented Requirements with UML</i>
Tipo de Publicação	Artigo de Conferência
Autores	ARAÚJO, João; MOREIRA, Ana; BRITO, Isabel; RASHID, Awais
Palavras-Chave	-
Ano de Publicação	2002
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling With UML (AOM)</i>

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
Tag	Extensão UML: UMLAUT
Grau de Relevância	Médio
ID	Meunier2010
Título	<i>Aspect-Oriented Software Development in Practice: Tales from AOSD-Europe</i>
Tipo de Publicação	Artigo de Conferência
Autores	RASHID, Awais; COTTENIER, Thomas; GREENWOOD, Phil; CHITCHYAN, Ruzanna; MEUNIER, Regine; COELHO, Roberta; SÜDHOLT, Mario; JOOSEN, Wouter
Palavras-Chave	-
Ano de Publicação	2010
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD)</i>
Tag	RSL
Grau de Relevância	Baixo
ID	Rebello2014
Título	<i>AspectJML: Modular Specification and Runtime Checking for Crosscutting Contracts</i>
Tipo de Publicação	Artigo de Conferência
Autores	REBÊLO, Henrique; LEAVENS, Gary T.; BAGHERZADEH, Mehdi; RAJAN, Hriday; LIMA, Ricardo Massa Ferreira; ZIMMERMAN, Daniel M.; CORNÉLIO, Márcio; THÜM, Thomas
Palavras-Chave	AspectJ; AspectJML; <i>Design by Contract</i> ; JML; <i>Aspect-Oriented Programming</i> ; <i>Crosscutting Contracts</i>
Ano de Publicação	2014
Periódico	<i>International Conference on Modularity</i>
Tag	<i>Design by Contract</i>
Grau de Relevância	Médio
ID	Georg2003
Título	<i>Composing Aspect Models</i>
Tipo de Publicação	Artigo de Conferência
Autores	GEORG, Geri; FRANCE, Robert; RAY, Indrakshi
Palavras-Chave	-
Ano de Publicação	2003
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling With UML (AOM)</i>
Tag	AOM

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
Grau de Relevância	Médio
ID	Clark2001a
Título	<i>Composition Patterns: An Approach to Designing Reusable Aspects</i>
Tipo de Publicação	Artigo de Conferência
Autores	CLARKE, Siobhán N.; WALKER, Robert J.
Palavras-Chave	-
Ano de Publicação	2001
Periódico	<i>ACM/IEEE International Conference on Software Engineering - ICSE</i>
Tag	Extensão UML: Outras Técnicas
Grau de Relevância	Médio
ID	Magableh2014b
Título	<i>Comprehensive Aspectual UML Approach to Support AspectJ</i>
Tipo de Publicação	Artigo de Periódico
Autores	MAGABLEH, Aws A.; SHUKUR, Zarina; ALI, Noorazean Mohd
Palavras-Chave	-
Ano de Publicação	2014
Periódico	<i>The Scientific World Journal</i>
Tag	RSL
Grau de Relevância	Médio
ID	Skotiniotis2004
Título	<i>Cona: Aspects for Contracts and Contracts for Aspects</i>
Tipo de Publicação	Artigo de Conferência
Autores	SKOTINIOTIS, Therapon; LORENZ, David
Palavras-Chave	<i>Design by Contract</i>
Ano de Publicação	2004
Periódico	ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications
Tag	<i>Design by Contract</i> sobre AOP
Grau de Relevância	Alto
ID	Skotiniotis2004a
Título	<i>Conaj: Generating Contracts as Aspects</i>
Tipo de Publicação	Relatório
Autores	SKOTINIOTIS, Therapon; LORENZ, David
Palavras-Chave	<i>Design by Contract</i>

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
Ano de Publicação	2004
Periódico	College of Computer and Information Science
Tag	<i>Design by Contract</i> sobre AOP
Grau de Relevância	Alto
ID	Rebelo2014b
Título	<i>Contract Modularity in Design by Contract Languages</i>
Tipo de Publicação	Tese de Doutorado
Autores	REBÊLO, Henrique
Palavras-Chave	<i>Design by Contract</i>
Ano de Publicação	2014
Periódico	-
Tag	<i>Design by Contract</i> sobre AOP
Grau de Relevância	Alto
ID	Wampler2006
Título	<i>Contract4J for Design by Contract in Java: Design Pattern-Like Protocols and Aspect Interfaces</i>
Tipo de Publicação	Artigo de Conferência
Autores	WAMPLER, Dean
Palavras-Chave	<i>Design by Contract</i> sobre AOP
Ano de Publicação	2006
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling (AOM)</i>
Tag	<i>Design by Contract</i>
Grau de Relevância	Alto
ID	Dario2005
Título	<i>Design by Contract com JML</i>
Tipo de Publicação	Artigo de Conferência
Autores	DARIO, Dalton; GUERRERO, Serey
Palavras-Chave	<i>Design by Contract</i> ; JML
Ano de Publicação	2005
Periódico	Congresso da Sociedade Brasileira de Computação
Tag	<i>Design by Contract</i>
Grau de Relevância	Alto
ID	Molderez2012
Título	<i>Design by Contract for Aspects, by Aspects</i>
Tipo de Publicação	Artigo de Conferência

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
Autores	MOLDEREZ, Tim; JANSSENS, Dirk
Palavras-Chave	<i>Design by Contract; Aspect-Oriented Programming</i>
Ano de Publicação	2012
Periódico	<i>Workshop on Foundations of Aspect-Oriented Languages - FOAL</i>
Tag	<i>Design by Contract sobre AOP</i>
Grau de Relevância	Alto
ID	Leavens2006
Título	<i>Design by Contract with JML</i>
Tipo de Publicação	<i>Draft Paper</i>
Autores	LEAVENS, Gary T.; CHEON, Yoonsik
Palavras-Chave	<i>Design by Contract; JML</i>
Ano de Publicação	2006
Periódico	-
Tag	<i>Design by Contract</i>
Grau de Relevância	Alto
ID	Mostefaoui2007
Título	<i>Design-Level Detection of Interactions in Aspect-UML Models Using Alloy</i>
Tipo de Publicação	Artigo de Periódico
Autores	MOSTEFAOUI, Farida; VACHON, Julie
Palavras-Chave	-
Ano de Publicação	2007
Periódico	<i>Journal of Object Technology</i>
Tag	Extensão UML: Outras Técnicas
Grau de Relevância	Alto
ID	Fuentes2007a
Título	<i>Designing and Weaving Aspect-Oriented Executable UML Models</i>
Tipo de Publicação	Artigo de Periódico
Autores	FUENTES, Lidia; SÁNCHEZ, Pablo
Palavras-Chave	-
Ano de Publicação	2007
Periódico	<i>Journal of Object Technology</i>
Tag	Extensão UML: WEAVR
Grau de Relevância	Alto
ID	Stein2002a
Título	<i>Designing Aspect-Oriented Crosscutting in UML</i>

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
Tipo de Publicação	Artigo de Conferência
Autores	STEIN, Dominik; HANENBERG, Stefan; UNLAND, Rainer
Palavras-Chave	-
Ano de Publicação	2002
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling With UML (AOM)</i>
Tag	Extensão UML: AODM
Grau de Relevância	Alto
ID	Ali2014
Título	<i>Does Aspect-Oriented Modeling Help Improve the Readability of UML State Machines?</i>
Tipo de Publicação	Livro
Autores	ALI, Shaukat; YUE, Tao; BRIAND, Lionel C.
Palavras-Chave	<i>Aspect-Oriented Modeling; Comprehension; Controlled Experiment; Defect Identification and Fixing; UML State Machines</i>
Ano de Publicação	2014
Periódico	<i>Software and Systems Modeling</i>
Tag	AOM
Grau de Relevância	Baixo
ID	Fuentes2006
Título	<i>Elaborating UML 2.0 Profiles for AO Design</i>
Tipo de Publicação	Artigo de Conferência
Autores	FUENTES, Pablo Sánchez Lidia
Palavras-Chave	-
Ano de Publicação	2006
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling (AOM)</i>
Tag	Extensão UML: AODM
Grau de Relevância	Alto
ID	Gowri2012
Título	<i>Extending the UML Metamodel to Grant Prop Up For Crosscutting Concerns</i>
Tipo de Publicação	Artigo de Periódico
Autores	GOWRI, Veluru
Palavras-Chave	<i>UML Metamodel; Aspect-Oriented Modeling; Modularity; Software Maintenance</i>

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
Ano de Publicação	2012
Periódico	<i>International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)</i>
Tag	AOM
Grau de Relevância	Médio
ID	Lorenz2004
Título	<i>Extending Design by Contract for Aspect-Oriented Programming</i>
Tipo de Publicação	Artigo de Conferência
Autores	LORENZ, David; SKOTINIOTIS, Therapon
Palavras-Chave	<i>Design by Contract;</i>
Ano de Publicação	2004
Periódico	<i>ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications</i>
Tag	<i>Design by Contract sobre AOP</i>
Grau de Relevância	Alto
ID	Sharafi2010
Título	<i>Extending the UML Metamodel to Provide Support for Crosscutting Concerns</i>
Tipo de Publicação	Artigo de Conferência
Autores	SHARAFI, Zohreh; MIRSHAMS, Parisa; HAMOU-LHADJ, Abdelwahab; CONSTANTINIDES, Constantinos
Palavras-Chave	<i>Aspect-Oriented Modeling; Modularity; Software Maintenance; UML Metamodel</i>
Ano de Publicação	2010
Periódico	<i>ACIS International Conference on Software Engineering Research, Management and Applications - SERA</i>
Tag	Extensão UML: Outras Técnicas
Grau de Relevância	Médio
ID	Suzuki1999
Título	<i>Extending UML with Aspects: Aspect Support in the Design Phase</i>
Tipo de Publicação	Artigo de Conferência
Autores	SUZUKI, Junichi; YAMAMOTO, Yoshikazu
Palavras-Chave	-
Ano de Publicação	1999

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
Periódico	<i>European Conference on Object-Oriented Programming (ECOOP) - Workshop on Object-Oriented Technology</i>
Tag	Extensão UML: Outras Técnicas
Grau de Relevância	Alto
ID	Guessi2011
Título	<i>Extensions of UML to Model Aspect-Oriented Software Systems</i>
Tipo de Publicação	Artigo de Periódico
Autores	GUESSI, Milena; OLIVEIRA, Lucas Bueno Ruas; NAKAGAWA, Elisa Yumi
Palavras-Chave	<i>Aspect-Orientation; Systematic Mapping; UML</i>
Ano de Publicação	2011
Periódico	<i>CLEI Electronic Journal</i>
Tag	RSL
Grau de Relevância	Baixo
ID	Kande2002a
Título	<i>From AOP to UML - A Bottom-Up Approach</i>
Tipo de Publicação	Artigo de Conferência
Autores	KANDÉ, Mohamed Mancona; KIENZLE, Jörg; STROHMEIER, Alfred
Palavras-Chave	-
Ano de Publicação	2002
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling With UML (AOM)</i>
Tag	Implementação: AspectJ
Grau de Relevância	Baixo
ID	Kande2002
Título	<i>From AOP to UML : Towards an Aspect-Oriented Architectural Modeling Approach</i>
Tipo de Publicação	Artigo de Conferência
Autores	KANDÉ, Mohamed Mancona; KIENZLE, Jörg; STROHMEIER, Alfred
Palavras-Chave	-
Ano de Publicação	2002
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling With UML (AOM)</i>
Tag	Implementação: AspectJ

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
Grau de Relevância	Médio
ID	Groher2003
Título	<i>Generating Aspect Code from UML Models</i>
Tipo de Publicação	Artigo de Conferência
Autores	GROHER, Iris; SCHULZE, Stefan
Palavras-Chave	AOSD; Aspect; AspectJ; UML; <i>Crosscutting Concerns</i>
Ano de Publicação	2003
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling With UML (AOM)</i>
Tag	Geração de Código
Grau de Relevância	Médio
ID	Kramer1998
Título	<i>iContract The Java Design by Contract Tool</i>
Tipo de Publicação	Artigo de Conferência
Autores	KRAMER, Reto
Palavras-Chave	<i>Design by Contract</i>
Ano de Publicação	1998
Periódico	<i>International Conference on Technology of Object-Oriented Languages and Systems - TOOLS</i>
Tag	<i>Design by Contract</i>
Grau de Relevância	Alto
ID	Basch2003
Título	<i>Incorporating Aspects into the UML</i>
Tipo de Publicação	Artigo de Conferência
Autores	BASCH, Mark; SANCHEZ, Arturo
Palavras-Chave	<i>Modeling Aspects; Aspect-Oriented Programming; UML</i>
Ano de Publicação	2003
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling With UML (AOM)</i>
Tag	Extensão UML: AODM
Grau de Relevância	Alto
ID	Sullivan2005
Título	<i>Information Hiding Interfaces for Aspect-Oriented Design</i>
Tipo de Publicação	Artigo de Conferência

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
Autores	SULLIVAN, Kevin; GRISWOLD, William G.; SONG, Yuanyuan; CAI, Yuanfang; SHONIE, Macneil; TEWARI, Nishit; RAJAN, Hri-desh
Palavras-Chave	<i>Aspect-Oriented Programming; Design Rules; Options</i>
Ano de Publicação	2005
Periódico	<i>European Software Engineering Conference and The ACM SIG-SOFT International Symposium on Foundations of Software Engineering - ESEC/FSE</i>
Tag	AOM
Grau de Relevância	Alto
ID	Karaorman1999
Título	<i>jContractor: A Reflective Java Library to Support Design By Contract</i>
Tipo de Publicação	Artigo de Conferência
Autores	KARAORMAN, Murat; BRUNO, John
Palavras-Chave	-
Ano de Publicação	1999
Periódico	<i>International Conference on Metalevel Architectures and Reflection</i>
Tag	<i>Design by Contract</i>
Grau de Relevância	Alto
ID	Abercrombie2002
Título	<i>jContractor: Bytecode Instrumentation Techniques for Implementing Design by Contract in Java</i>
Tipo de Publicação	Artigo de Periódico
Autores	ABERCROMBIE, Parker; KARAORMAN, Murat
Palavras-Chave	-
Ano de Publicação	2002
Periódico	<i>Electronic Notes in Theoretical Computer Science</i>
Tag	<i>Design by Contract</i>
Grau de Relevância	Alto
ID	Leavens2013
Título	<i>JML Reference Manual</i>
Tipo de Publicação	Manual
Autores	LEAVENS, Gary T.; POLL, Erik; CLIFTON, Curtis; CHEON, Yo-onsik; RUBY, Clyde

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
Palavras-Chave	-
Ano de Publicação	2013
Periódico	-
Tag	<i>Design by Contract</i>
Grau de Relevância	Alto
ID	Feldman2006
Título	<i>Jose: Aspects for Design by Contract</i>
Tipo de Publicação	Manual
Autores	FELDMAN, Yishai; BARZILAY, Ohad; TYSZBEROWICZ, Shmuel
Palavras-Chave	-
Ano de Publicação	2006
Periódico	<i>Workshop on Formal Techniques for Java Programs - FTJP</i>
Tag	<i>Design by Contract sobre AOP</i>
Grau de Relevância	Alto
ID	Medeiros2008
Título	MaRiSa-D: Uma Abordagem Baseada em Modelos para Desenvolvimento de Software Orientado a Aspectos
Tipo de Publicação	Artigo de Conferência
Autores	GOTTARDI, Thiago; CAMARGO, Valter Vieira de
Palavras-Chave	-
Ano de Publicação	2008
Periódico	<i>Latin American Workshop on Aspect-Oriented Software Development - LA-WASP</i>
Tag	Extensão UML: aSideML
Grau de Relevância	Alto
ID	Cottenier2006
Título	<i>Modeling Aspect-Oriented Compositions</i>
Tipo de Publicação	Artigo de Conferência
Autores	COTTENIER, Thomas; VAN DEN BERG, Aswin; ELRAD, Tzilla
Palavras-Chave	<i>Aspect Oriented Modeling; Model-Driven Architecture; Model-Driven Development; UML 2</i>
Ano de Publicação	2006
Periódico	<i>ACM/IEEE International Conference on Model-Driven Engineering Languages and Systems - MODELS</i>
Tag	Extensão UML: AODM
Grau de Relevância	Médio

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
ID	Bustos2007
Título	<i>Modeling Aspects With UML's Class, Sequence and State Diagrams in an Industrial Setting</i>
Tipo de Publicação	Artigo de Conferência
Autores	BUSTOS, Alex; ETEROVIC, Yadran
Palavras-Chave	<i>Aspect Orientation; Software Design; UML</i>
Ano de Publicação	2007
Periódico	<i>IASTED International Conference on Software Engineering and Applications - SEA</i>
Tag	Extensão UML: AODM
Grau de Relevância	Médio
ID	Coelho2004
Título	<i>Modeling Aspects: An Implementation-Driven Approach</i>
Tipo de Publicação	Artigo de Conferência
Autores	COELHO, Wesley; MURPHY, Gail C.
Palavras-Chave	-
Ano de Publicação	2004
Periódico	<i>Best Practices for Model-driven Software Development Workshop at OOPSLA 2004</i>
Tag	Extensão UML: AODM
Grau de Relevância	Alto
ID	Cooper2003
Título	<i>Modeling Performance as an Aspect : a UML Based Approach</i>
Tipo de Publicação	Artigo de Conferência
Autores	COOPER, Kendra; DAI, Lirong; DENG, Yi; DONG, Jing
Palavras-Chave	<i>Aspect-Oriented; Formal Methods; Rapide; UML</i>
Ano de Publicação	2003
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling With UML (AOM)</i>
Tag	AOM
Grau de Relevância	Baixo
ID	Stein2004
Título	<i>Modeling Pointcuts</i>
Tipo de Publicação	Artigo de Periódico
Autores	STEIN, Dominik; HANENBERG, Stefan; UNLAND, Rainer
Palavras-Chave	-

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
Ano de Publicação Periódico Tag Grau de Relevância	2004 <i>Conference on Aspect-Oriented Software Development (AOSD) – Early Aspect Workshop (EA) – Aspect-Oriented Requirements Engineering and Architecture Design</i> Extensão UML: AODM Alto
ID Título Tipo de Publicação Autores Palavras-Chave Ano de Publicação Periódico Tag Grau de Relevância	Ali2012a <i>Modeling Robustness Behavior Using Aspect-Oriented Modeling to Support Robustness Testing of Industrial Systems</i> Artigo de Periódico ALI, Shaukat; BRIAND, Lionel C.; HEMMATI, Hadi <i>Aspect-Oriented Modeling; Crosscutting Behavior; Robustness; Robustness Testing; UML Profile; UML State Machines</i> 2012 <i>Software and Systems Modeling</i> AOM Baixo
ID Título Tipo de Publicação Autores Palavras-Chave Ano de Publicação Periódico Tag Grau de Relevância	Sullivan2010 <i>Modular Aspect-Oriented Design with XPIs</i> Artigo de Periódico SULLIVAN, Kevin; GRISWOLD, William G.; RAJAN, Hridesh; SONG, Yuanyuan; CAI, Yuanfang; SHONIE, Macneil; TEWARI, Nishit <i>Aspect-Oriented Programming; Design; Design Rules; Options</i> 2010 <i>ACM Transactions on Software Engineering and Methodology - TOSEM</i> AOM Baixo
ID Título Tipo de Publicação Autores	Rebelo2014a <i>Modularizing Crosscutting Contracts with AspectJML</i> Artigo de Conferência REBÊLO, Henrique; LEAVENS, Gary T.; BAGHERZADEH, Mehdi; RAJAN, Hridesh; LIMA, Ricardo Massa Ferreira; ZIMMERMAN, Daniel M.; CORNÉLIO, Márcio; THÜM, Thomas

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
Palavras-Chave	AspectJ; AspectJML; <i>Design by Contract</i> ; JML; <i>Permission</i> ; <i>Aspect-Oriented Programming</i> ; <i>Crosscutting Contracts</i>
Ano de Publicação	2014
Periódico	<i>International Conference on Modularity</i>
Tag	<i>Design by Contract</i>
Grau de Relevância	Médio
ID	Cottenier2007
Título	<i>Motorola WEAVR: Aspect Orientation and Model-Driven Engineering</i>
Tipo de Publicação	Artigo de Periódico
Autores	COTTENIER, Thomas; VAN DEN BERG, Aswin; ELRAD, Tzilla
Palavras-Chave	-
Ano de Publicação	2007
Periódico	<i>Journal of Object Technology</i>
Tag	Extensão UML: WEAVR
Grau de Relevância	Alto
ID	Meyer2000
Título	<i>Object-Oriented Software Construction</i>
Tipo de Publicação	Livro
Autores	MEYER, Bertrand
Palavras-Chave	-
Ano de Publicação	2000
Periódico	-
Tag	Extensão <i>Design by Contract</i>
Grau de Relevância	Alto
ID	Coelho2011
Título	<i>On the Interplay of Exception Handling and Design by Contract: An Aspect-Oriented Recovery Approach</i>
Tipo de Publicação	Artigo de Conferência
Autores	COELHO, Roberta; LIMA, Ricardo; LEAVENS Gary T.; HUISMAN, Marieke; MOTA, Alexandre; CASTOR, Fernando
Palavras-Chave	-
Ano de Publicação	2011
Periódico	<i>Workshop on Formal Techniques for Java Programs - FTJP</i>
Tag	Extensão <i>Design by Contract</i>
Grau de Relevância	Alto

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
ID	Woodside2009
Título	<i>Performance Analysis of Security Aspects by Weaving Scenarios Extracted from UML Models</i>
Tipo de Publicação	Artigo de Periódico
Autores	WOODSIDE, C. Murray; PETRIU, Dorina C.; PETRIU, Dorin B.; XU, Jing; ISRAR, Tauseef; GEORG, Geri; FRANCE, Robert; BIEMAN, James M.; HOUMB, Siv Hilde; JÜRJENS, Jan
Palavras-Chave	<i>Aspect-Oriented Modeling; Software Performance Engineering</i>
Ano de Publicação	2009
Periódico	<i>The Journal of Systems & Software</i>
Tag	AOM
Grau de Relevância	Baixo
ID	Petriu2007
Título	<i>Performance Analysis of Security Aspects in UML Models</i>
Tipo de Publicação	Artigo de Conferência
Autores	PETRIU, Dorina C.; WOODSIDE, C. Murray; PETRIU, Dorin B.; XU, Jing; ISRAR, Tauseef; GEORG, Geri; FRANCE, Robert; BIEMAN, James M.; HOUMB, Siv Hilde; JÜRJENS, Jan
Palavras-Chave	<i>Aspect-Oriented Modeling; Model Transformations; Security; Software Performance Engineering; UML</i>
Ano de Publicação	2007
Periódico	<i>International Workshop on Software and Performance - WOSP</i>
Tag	AOM
Grau de Relevância	Baixo
ID	Zhao2003
Título	Pipa: A Behavioral Interface Specification Language for AspectJ
Tipo de Publicação	Artigo de Conferência
Autores	ZHAO, Jianjun; RINARD, Martin
Palavras-Chave	<i>Design by Contract</i>
Ano de Publicação	2003
Periódico	<i>International Conference on Fundamental Approaches to Software Engineering (FASE)</i>
Tag	<i>Design by Contract sobre AOP</i>
Grau de Relevância	Alto
ID	Silva2007
Título	Refatoração de Modelos Orientados a Aspectos

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
Tipo de Publicação	Artigo de Conferência
Autores	SILVA, Bruno Carreiro da; NUNES, Daltro José
Palavras-Chave	-
Ano de Publicação	2007
Periódico	<i>Workshop de Teses e Dissertações em Engenharia de Software</i>
Tag	AOM
Grau de Relevância	Alto
ID	Cepa2005
Título	Representing Explicit Attributes in UML
Autores	CEPA, Vasian; KLOPPENBURG, Sven
Palavras-Chave	-
Ano de Publicação	2005
Periódico	<i>Workshop on Formal Techniques for Java Programs - FTJP</i>
Tag	<i>Design by Contract</i>
Grau de Relevância	Alto
ID	Cottenier2007b
Título	<i>Stateful Aspects: The Case for Aspect-oriented Modeling</i>
Tipo de Publicação	Artigo de Conferência
Autores	COTTENIER, Thomas; VAN DEN BERG, Aswin; ELRAD, Tzilla
Palavras-Chave	<i>Aspect-Oriented Modeling; Model-Driven Software Development; State Machine Diagrams</i>
Ano de Publicação	2007
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling (AOM)</i>
Tag	Extensão UML: WEAVR
Grau de Relevância	Médio
ID	Magableh2012
Título	<i>Systematic Review on Aspect-Oriented UML Modeling: A Complete Aspectual UML Modeling Framework</i>
Tipo de Publicação	Artigo de Periódico
Autores	MAGABLEH, Aws A.; SHUKUR, Zarina; ALI, Noorazeen Mohd
Palavras-Chave	<i>Aspectual UML Diagrams; Aspect Modelling; Aspect Modularization; Aspect Representations; Aspect-Oriented UML; Aspects</i>
Ano de Publicação	2012
Periódico	<i>Journal of Applied Sciences</i>
Tag	RSL

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
Grau de Relevância	Médio
ID	Chibani2013
Título	<i>Towards a New Aspect-Oriented Modelling Approach</i>
Tipo de Publicação	Artigo de Periódico
Autores	CHIBANI, Meriem; BELATTAR, Brahim; BOUROUIS, Abdelhabib
Palavras-Chave	<i>Aspect Oriented Programming; UML Profile; Aspect Oriented Software Development; Aspect Oriented Modelling; Crosscutting Concerns; Separation of Concerns</i>
Ano de Publicação	2013
Periódico	<i>Conference on Theoretical and Applicative Aspects of Computer Science - ctAAcs</i>
Tag	AOM
Grau de Relevância	Médio
ID	Chibani2013a
Título	<i>Towards a UML Meta Model Extension for Aspect-Oriented Modeling</i>
Tipo de Publicação	Artigo de Conferência
Autores	CHIBANI, Meriem; BELATTAR, Brahim; BOUROUIS, Abdelhabib
Palavras-Chave	<i>Aspect-Oriented Programming (AOP); Aspect-Oriented Software Development; AspectJ; UML; Profile</i>
Ano de Publicação	2013
Periódico	<i>International Conference on Software Engineering Advances - ICSEA</i>
Tag	AOM
Grau de Relevância	Médio
ID	Krechetov2006
Título	<i>Towards an Integrated Aspect-Oriented Modeling Approach for Software Architecture Design</i>
Tipo de Publicação	Artigo de Conferência
Autores	KRECHETOV, Ivan; TEKINERDOGAN, Bedir; GARCIA, Alessandro; CHAVEZ, Christina Von Flach; KULESZA, Uirá
Palavras-Chave	<i>Aspect-Oriented Software Architecture; UML; Visual Modeling</i>
Ano de Publicação	2006
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling (AOM)</i>
Tag	Extensão UML: aSideML

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
Grau de Relevância	Alto
ID	Ozkaya2013
Título	<i>Towards Design-by-Contract Based Software Architecture Design</i>
Tipo de Publicação	Artigo de Periódico
Autores	OZKAYA, Mert; KLOUKINAS, Christos
Palavras-Chave	-
Ano de Publicação	2013
Periódico	<i>Intelligent Software Methodologies, Tools and Techniques - SoMeT</i>
Tag	<i>Design by Contract</i>
Grau de Relevância	Alto
ID	Fuentes2007b
Título	<i>Towards Executable Aspect-Oriented UML Models</i>
Tipo de Publicação	Artigo de Conferência
Autores	FUENTES, Lidia; SÁNCHEZ, Pablo
Palavras-Chave	<i>Aspect-Oriented Modeling; Aspect-Oriented ModelWeaver; Aspect-Oriented Design Languages; Aspect-Oriented UML 2.0 Profile</i>
Ano de Publicação	2007
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling (AOM)</i>
Tag	Extensão UML: WEAVR
Grau de Relevância	Alto
ID	Chavez2004
Título	Um Enfoque Baseado em Modelos para o Design Orientado a Aspectos
Tipo de Publicação	Tese de Doutorado
Autores	CHAVEZ, Christina Von Flach
Palavras-Chave	<i>Aspect-Oriented Software Development; Aspect-Oriented Languages; Aspect Model; Metamodeling; Separation of Concerns</i>
Ano de Publicação	2004
Periódico	-
Tag	Extensão UML: aSideML
Grau de Relevância	Alto
ID	UetanabaraJunior2006
Título	Um Perfil UML para Projeto de Frameworks Transversais

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
Tipo de Publicação	Artigo de Conferência
Autores	UETANABARA JÚNIOR, José; CAMARGO, Valter Vieira de
Palavras-Chave	-
Ano de Publicação	2006
Periódico	<i>Latin American Workshop on Aspect-Oriented Software Development - LA-WASP</i>
Tag	Extensão UML: Outras Técnicas
Grau de Relevância	Médio
ID	Guimaraes2009
Título	Uma Estratégia Baseada em Metamodelo para Geração de Código Orientado a Aspectos
Tipo de Publicação	Artigo de Conferência
Autores	GUIMARÃES, Everton T; CACHO, Nélio; BATISTA, Thais V.
Palavras-Chave	-
Ano de Publicação	2009
Periódico	Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software - SBCARS
Tag	Geração de Código
Grau de Relevância	Baixo
ID	Losavio2009
Título	<i>UML Extensions for Aspect Oriented Software Development</i>
Tipo de Publicação	Artigo de Periódico
Autores	LOSAVIO, Francisca; MATTEO, Alfredo; MORANTES, Patricia
Palavras-Chave	-
Ano de Publicação	2009
Periódico	<i>Journal of Object Technology</i>
Tag	RSL
Grau de Relevância	Alto
ID	Aldawud2003
Título	<i>UML Profile for Aspect-Oriented Software Development</i>
Tipo de Publicação	Artigo de Conferência
Autores	ALDAWUD, Omar; ELRAD, Tzilla; BADER, Atef
Palavras-Chave	-
Ano de Publicação	2003
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling with UML (AOM)</i>

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
Tag	Extensão UML: AODM
Grau de Relevância	Alto
ID	UetanabaraJunior2009
Título	<i>UML-AOF: A Profile for Modeling Aspect-Oriented Frameworks</i>
Tipo de Publicação	Artigo de Conferência
Autores	UETANABARA JÚNIOR, José; CAMARGO, Valter Vieira de; CHAVEZ, Christina Von Flach
Palavras-Chave	<i>Aspect-Oriented Frameworks; Aspect-Oriented Modeling</i>
Ano de Publicação	2009
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling (AOM)</i>
Tag	Extensão UML: aSideML
Grau de Relevância	Alto
ID	Ho2000
Título	<i>UMLAUT: A Framework for Weaving UML-Based Aspect-Oriented Designs</i>
Tipo de Publicação	Artigo de Periódico
Autores	HO, Wai-Ming; PENNANEAC'H, François; PLOUZEAU, Noël
Palavras-Chave	-
Ano de Publicação	2000
Periódico	<i>International Conference on Technology of Object-Oriented Languages and Systems - TOOLS</i>
Tag	Extensão UML: UMLAUT
Grau de Relevância	Alto
ID	Mouheb2009
Título	<i>Weaving Security Aspects Into UML 2.0 Design Models</i>
Tipo de Publicação	Artigo de Periódico
Autores	MOUHEB, Djedjiga; TALHI, Chamseddine; LIMA, Vitor; DEBBABI, Mourad; WANG, Lingyu; POURZANDI, Makan
Palavras-Chave	<i>Design; Security</i>
Ano de Publicação	2009
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling (AOM)</i>
Tag	AOM
Grau de Relevância	Baixo
ID	Groher2007

Continua na próxima página.

Quadro 12 - Listagem das Publicações Seleccionadas pela RSL (continuação).

Atributo	Publicação
Título	<i>XWeave: Models and Aspects in Concert</i>
Tipo de Publicação	Artigo de Conferência
Autores	GROHER, Iris; VOELTER, Markus
Palavras-Chave	<i>Aspect-Oriented Software Development; Model Weaving; Model-Driven Software Development</i>
Ano de Publicação	2007
Periódico	<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling (AOM)</i>
Tag	Extensão UML: AODM
Grau de Relevância	Alto

Fonte: Desenvolvido pelo autor.

**APÊNDICE D – LISTAGEM DOS PERIÓDICOS E MEIOS DE PUBLICAÇÃO
SELECIONADOS PELA RSL**

Tabela 11 – Relação dos Meios de Publicações dos Periódicos

Periódico	Publicação
<i>ACIS International Conference on Software Engineering Research, Management and Applications - SERA</i>	1
<i>ACIS International Journal of Computer Information Science</i>	1
<i>ACM International C* Conference on Computer Science & Software Engineering - C3S2E</i>	1
<i>ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages and Application</i>	2
<i>ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages and Application - OOPSLA - Workshop on Aspect Oriented Programming</i>	1
<i>ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages and Application - OOPSLA - Workshop on Best Practices for Model-driven Software Development</i>	1
<i>ACM Transactions on Software Engineering and Methodology – TOSEM</i>	1
<i>ACM/IEEE International Conference on Model-Driven Engineering Languages and Systems - MODELS</i>	1
<i>ACM/IEEE International Conference on Software Engineering - ICSE</i>	1
<i>CLEI Electronic Journal</i>	1
<i>Computing and Informatics</i>	1
<i>Conference of the Center for Advanced Studies on Collaborative Research - CASCON</i>	1
<i>Conference on Aspect-Oriented Software Development (AOSD)</i>	4
<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling (AOM)</i>	11

Continua na próxima página.

Tabela 11 - Meios de Publicação dos Periódicos - (continuação).

Periódico	Publicação
<i>Conference on Aspect-Oriented Software Development (AOSD) - Workshop on Aspect-Oriented Modeling With UML (AOM)</i>	16
<i>Conference on Aspect-Oriented Software Development (AOSD) – Early Aspect Workshop (EA) – Aspect-Oriented Requirements Engineering and Architecture Design</i>	1
<i>Conference on Theoretical and Applicative Aspects of Computer Science - ctAAcs</i>	1
<i>Congresso da Sociedade Brasileira de Computação</i>	1
<i>Draft Paper</i>	1
<i>Electronic Notes in Theoretical Computer Science</i>	1
<i>European Conference on Object-Oriented Programming (ECOOP) - Workshop on Object-Oriented Technology</i>	1
<i>European Software Engineering Conference and The ACM SIGSOFT International Symposium on Foundations of Software Engineering - ESEC/FSE</i>	1
<i>IASTED International Conference on Software Engineering and Applications - SEA</i>	1
<i>IEEE International Enterprise Distributed Object Computing Workshop - EDOC</i>	1
<i>Information and Software Technology</i>	1
<i>Intelligent Software Methodologies, Tools and Techniques – SoMeT</i>	1
<i>International Conference on Fundamental Approaches to Software Engineering (FASE)</i>	1
<i>International Conference on Metalevel Architectures and Reflection</i>	1
<i>International Conference on Modularity</i>	2
<i>International Conference on Software Engineering Advances - ICSEA</i>	1
<i>International Conference on Technology of Object-Oriented Languages and Systems - TOOLS</i>	2

Continua na próxima página.

Tabela 11 - Meios de Publicação dos Periódicos - (continuação).

Periódico	Publicação
<i>International Conference on UML Modeling Languages and Applications</i>	1
<i>International Journal of Advanced Research in Computer Engineering & Technology – IJAR CET</i>	1
<i>International Journal of Intelligent Systems</i>	1
<i>International Workshop on Early Aspects at ICSE</i>	1
<i>International Workshop on Software and Performance - WOSP</i>	1
<i>Journal ACM Computing Surveys</i>	1
<i>Journal of Applied Sciences</i>	2
<i>Journal of Object Technology</i>	6
<i>Latin American Workshop on Aspect-Oriented Software Development - LA-WASP</i>	2
<i>Lecture Notes in Computer Science</i>	1
<i>Livro</i>	1
<i>Manual de Referência</i>	1
<i>Report of College of Computer and Information Science</i>	1
<i>Report of Vienna University of Technology</i>	1
<i>Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software - SBCARS</i>	1
<i>Software and Systems Modeling</i>	2
<i>Tese da Pontifícia Universidade Católica do Rio de Janeiro</i>	1
<i>Tese da Universidade Federal de Pernambuco</i>	1
<i>The Journal of Systems & Software</i>	1
<i>The Scientific World Journal</i>	1
<i>Workshop de Teses e Dissertações em Engenharia de Software</i>	1
<i>Workshop on Foundations of Aspect-Oriented Languages – FOAL</i>	1

Continua na próxima página.

Tabela 11 - Meios de Publicação dos Periódicos - (continuação).

Periódico	Publicação
<i>Workshop on Formal Techniques for Java Programs – FTJP</i>	2
Total	89

Fonte: Desenvolvido pelo autor.

ANEXO A – PRIMEIRO PARECER TÉCNICO

PARECER TÉCNICO

Exame do artefato como solução da Dissertação de Mestrado Profissional intitulada “Uma Extensão do Diagrama de Classe em *Aspect-Oriented Design Model* (AODM) para modelagem de *Design by Contract*” apresentada por Fabrício Soares da Silva ao Programa de Pós-Graduação *stricto sensu* do Mestrado Profissional em Sistemas de Informação da Universidade Federal de Pernambuco sob orientação do Prof. Dr Henrique Emanuel Mostaert Rebêlo.

A dissertação do mestrando Fabrício Soares da Silva, Uma Extensão do Diagrama de Classe em *Aspect-Oriented Design Model* (AODM) para modelagem de *Design by Contract*, trata do problema à respeito da falta de uma técnica de modelagem de DbC (*Design by Contract*) ou Programação por Contrato, que prejudica as equipes de desenvolvimento de *software* por não existir uma representação gráfica do sistema, para orientar o desenvolvimento e a solução de artefatos que utilizam esta técnica. O código do programa é a única documentação com as regras que contemplam o Aspecto atualmente.

A técnica DbC (*Design by Contract*) ou Programação por Contrato é uma técnica criada por Meyer (2002) definida pela sua aderência à OO fundamentadas por regras, ou "contratos" para uso dos métodos, de modo a garantir a confiabilidade do sistema a ser desenvolvido. Tal técnica foi implementada no JML (*Java Modeling Language*) ou Linguagem de Modelagem para Java, que é uma linguagem formal de especificação de interface comportamental para Java, com notações importantes usadas no DbC como subconjunto próprio. Existem também alguns exemplos de aplicações DbC que aplicam contratos apontados por Meyer (2002) em elementos do Aspecto que são: AspectJML, Barter, ConaJ, Contract4J, ContractAJ, Jose e Pipa. O autor descreve com muitos embasamentos teóricos sobre essas definições e aplicações do uso do DbC e com muita propriedade sobre o encadeamento de sua proposta, bem como a relevância de um

estudo aprofundado sobre o tema, que contribui para uma exigência do mercado de desenvolvimento de *software*, pois ainda está deficiente com literaturas que tratam do assunto e incorporam exemplos práticos.

A apresentação e desenvolvimento do projeto do artefato tratam diretamente da questão de pesquisa, com as seguintes abordagens técnicas: AODM (*Aspect Oriented Design Model*), que embasa o Diagrama de Classes com Aspectos e a sintaxe AspectJML para exemplificar a apresentação desse projeto. O exemplo apresentado foi uma classe simples “ContaCorrente”, utilizando a linguagem de programação Java. O exemplo do aspecto foi chamado de AspectLogConta-Corrente, de maneira análoga, utilizou a linguagem de programação Java.

A exemplificação do artefato foi mostrada, utilizando esses exemplos para representar os contratos DbC e compor o Diagrama de Classe, de acordo com AODM. O autor descreve de forma bem aprofundada sobre essas técnicas utilizadas. A apresentação e a exemplificação do artefato são realizadas de modo didático, com o objetivo de facilitar o entendimento da proposta de solução do problema de pesquisa.

Para finalizar, o autor avaliou o artefato através do Diagrama de Classe AODM com suporte a contratos DbC para fundamentar o modelo proposto nesta pesquisa, o modelo AODM estendido, que combina todas as técnicas e as abordagens apresentadas ao longo da dissertação.

Concluo, portanto, que o artefato foi desenvolvido, utilizando o encadeamento proposto pela metodologia DSR, seguindo o rigor científico. Parabênizo o autor pelo trabalho de excelência, por fazer uma discussão aprofundada de um assunto importante para a comunidade de desenvolvimento de *software*.

Rio de Janeiro, 09 de novembro de 2017



Soraia Pacheco de Almeida Silva Felício

Doutora em História das Ciências e das Técnicas e Epistemologia (HCTE/UFRJ)
Mestre em Informática pelo Programa de Pós-Graduação em Informática (PPGI/NCE/UFRJ)

ANEXO B – SEGUNDO PARECER TÉCNICO

Parecer

Exame do Artefato descrito no Capítulo quatro da dissertação de Mestrado Profissional intitulada “Uma Extensão do Diagrama de Classe em Aspect-Oriented Design Model(AODM) para modelagem de Design by Contract” apresentada por Fabrício Soares da Silva ao Programa de Pós-Graduação stricto sensu da UFPE - Universidade Federal de Pernambuco sob orientação do Prof. Dr. Henrique Emanuel Mostaert Rebêlo.

Ao longo do capítulo quatro, intitulado “Projeto do Artefato” o mestrando disserta sobre as técnicas base para a apresentação do artefato, descrevendo o modelo AODM (Aspect Oriented Design Model) para estender conceitos de UML e fornecer conceitos orientados a aspecto; descrevendo a JML para implementar um modelo Design by Contract; e descrevendo o AspectJML que é uma extensão orientada a aspectos para a JML.

O mestrando almeja demonstrar que, combinando as técnicas acima, será possível expressar contratos em classes e aspectos, sem ferir os princípios da UML. Para tanto demonstra uma série de exemplos que apresentam a supracitada proposta e dão indícios de que a implementação é factível.

Para desenvolver o artefato o mestrando se vale da pesquisa de Cepa e Kloppenburg e opta por usar a apresentação por estereótipos que, em seu entender, se mostra mais adequada para ser adaptada para contratos DbC.

Ao avaliar a solução proposta o mestrando afirma que a implementação favorece a interpretação por ferramentas CASES, em função do seu suporte a estereótipos e propriedades, afirma também que a visibilidade do contrato é outro ponto positivo que facilita a leitura do artefato e a sua modelagem.

Parece claro que a proposta, ora exposta, apresenta resultados benfazejos e oferece várias alternativas para implementações futuras, podendo resultar em soluções de mercado para ferramentas de modelagem e geração de códigos automáticos por meio do Modelo AODM.

Rio de Janeiro, 07/11/2017



Prof. MSc. Luiz Francisco Dias Pereira - UFRJ
Diretor de Tecnologia da Informação
Pró-Reitoria de Planejamento e Desenvolvimento Institucional
Colégio Pedro II