

FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”  
CENTRO UNIVERSITÁRIO “EURÍPIDES DE MARÍLIA” – UNIVEM  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

**RAFAEL HENRIQUE DE MORAES AUGUSTO**

**UMA COMPARAÇÃO ENTRE APLICAÇÕES ORIENTADAS A  
OBJETOS E ORIENTADAS A  
ASPECTOS COM O USO DE MÉTRICAS**

MARÍLIA

2007

**RAFAEL HENRIQUE DE MORAES AUGUSTO**

**UMA COMPARAÇÃO ENTRE APLICAÇÕES ORIENTADAS A  
OBJETOS E ORIENTADAS A  
ASPECTOS COM O USO DE MÉTRICAS**

Monografia apresentada ao Curso de Ciência da Computação da Fundação de Ensino “Eurípides Soares da Rocha”. Mantenedora do Centro Universitário Eurípides de Marília – UNIVEM, com requisito parcial para obtenção do grau de Bacharel em Ciências da Computação.

Orientador:  
Prof. Dr. Valter Vieira de Camargo

MARÍLIA

2007

**RAFAEL HENRIQUE DE MORAES AUGUSTO**

**UMA COMPARAÇÃO ENTRE APLICAÇÕES ORIENTADAS A  
OBJETOS E ORIENTADAS A  
ASPECTOS COM O USO DE MÉTRICAS**

Banca examinadora da monografia apresentada ao Curso de Bacharelado em Ciência da Computação do “UNIVEM/F.E.E.S.R”, para obtenção do Título de Bacharel em Ciência da Computação.

Nota: \_\_\_\_\_(\_\_\_\_\_)

ORIENTADOR: Prof. Dr. Valter Vieira de Camargo.

1º EXAMINADOR: \_\_\_\_\_

2º EXAMINADOR: \_\_\_\_\_

Marília, \_\_\_\_ de \_\_\_\_\_ de 2007.

## **AGRADECIMENTOS**

“Que os nossos esforços desafiem as grandes impossibilidades, lembrai-vos, de que as grandes proezas da história, foram conquistadas do que parecia Impossível.”

(Charles Chaplin).

Aos meus pais, familiares e amigos queridos, que em todo momento colaboraram para a conclusão de um objetivo profissional.

Ao Prof. Dr. Valter Vieira de Camargo, que com muita sabedoria me conduziu à realização deste trabalho.

AUGUSTO, Rafael Henrique Moraes; **Uma comparação entre Aplicações Orientadas a Objetos e Orientadas a Aspectos com uso de Métricas**. 2007. Monografia (Conclusão de Curso em Ciência da Computação) – Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha, Marília, 2007.

## RESUMO

O Desenvolvimento de Software Orientado a Aspecto (DSOA) é um paradigma que tem como objetivo melhorar a modularização de um sistema de software e com isso melhorar sua manutenibilidade e reusabilidade. A qualidade do código é um requisito implícito que não é percebido pelo usuário, mas que afeta diretamente os desenvolvedores durante a manutenção e durante a criação de novas funcionalidades. Medir essa qualidade não é tarefa fácil. Para auxiliar nessa tarefa, estão sendo propostas nos últimos anos métricas de software, que avaliam os requisitos implícitos e regras heurísticas para avaliação dos resultados das métricas. Nessa monografia são utilizadas e algumas métricas já existente na literatura e também são propostas novas métricas e regras heurísticas. Essas métricas e regras heurísticas são aplicadas em dois softwares desenvolvidos anteriormente, ambos desenvolvidos nos paradigmas Orientado a Objetos e Orientado a Aspectos. Com isso é possível medir o nível de separação de interesses de cada um dos softwares. Como resultado, observou-se que o paradigma orientado a aspectos permite modularizar um software de forma mais adequada do que o paradigma orientado a objetos.

Palavras-chave: Métricas de software, desenvolvimento de software orientado a aspecto, regras heurísticas.

AUGUSTO, Rafael Henrique de Moraes; **Um estudo sobre Métricas de Comparação entre Sistemas Orientados a Objeto e Sistemas Orientados a Aspectos**. 2007. Monografia (Conclusão de Curso em Ciência da Computação) – Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha, Marília, 2007.

## ABSTRACT

The Development of the Aspect Oriented Software is a paradigm that recent that aims to improve the quality of the code and thereby improve their manutenibilidade and reusability of software, through better separation of concern. The quality of the code is a requirement implied that it is not perceived by the user, but it directly affects the developer, in time to manutenção and the creation of new functionality. Measuring the quality is not easy, to assist in that task, estão being proposed in the last years of software metrics, which evaluate the requirements implicit and are also being proposed rules heurísticas for evaluation of the results of the metrics. In this monograph are utilizadas and some metrics existing in the literature and proposed new metrics and heurísticas rules, such metrics and rules heurísticas are implemented in software CDSStore and Abacoo, both developed in the Object Oriented paradigm and Orientos the aspects . This will be possible to measure the results achieved by each of the paradigms and help the desenvolvedor the choice between them.

**Keywords:** Metrics of software, The Development of the Aspect Oriented Software, heurísticas rules.

## LISTA DE FIGURAS

Figura 1- Interface da Ferramenta AJATO.....24

## LISTA DE TABELAS

Tabela 1- Conjunto de Regras Heuristicas baseadas em uma única Métrica.....	21
Tabela 2- Conjunto de Regras Heuristicas de Separação de Interesse.....	22
Tabela 2.1- Valores e Limites utilizados nas Regras Heuristicas.....	23
Tabela 3- Conjunto de Regras Heuristicas de Sugeridas.....	27
Tabela 4- Aplicação da Métrica VS.....	29
Tabela 5- Aplicação da Métrica NOA.....	30
Tabela 6- Aplicação da Métrica NOO.....	30
Tabela 7- Aplicação da Métrica WOC.....	31
Tabela 8- Aplicação da Métrica LOC.....	31
Tabela 9- Aplicação da Métrica DIT.....	31
Tabela 10- Aplicação da Métrica CDC.....	32
Tabela 11- Aplicação da Métrica CDO.....	32
Tabela 12- Aplicação da Métrica NOAConcern.....	33
Tabela 13- Aplicação da Métrica NOOConcern.....	33
Tabela 14- Aplicação da Métrica LOCCConcern.....	34
Tabela 15- Aplicação da Métrica CDPDC.....	34
Tabela 16- Aplicação da Métrica CDPDO.....	35
Tabela 17- Aplicação da Métrica NOAConcernPDC.....	35
Tabela 18- Aplicação da Métrica CDTDC.....	36
Tabela 19- Aplicação da Métrica LOCCConcernPDC.....	36
Tabela 20- Aplicação da Métrica TNOAConcern.....	37
Tabela 21- Aplicação da Métrica TLOCCConcern.....	37
Tabela 22- Aplicação das Regras Heuristicas R01 e R02.....	55
Tabela 23- Aplicação das Regras Heuristicas R03 e R04.....	56
Tabela 24- Aplicação das Regras Heuristicas R05 e R06.....	58
Tabela 25- Aplicação das Regras Heuristicas R07 e R08.....	61

## LISTA DE GRÁFICOS

Gráfico 1- VS Ábaco .....	38
Gráfico 2- VS CDStore .....	38
Gráfico 3- NOA Ábaco .....	39
Gráfico 4- NOA CDStore .....	39
Gráfico 5- NOO Ábaco .....	40
Gráfico 6- NOO CDStore .....	40
Gráfico 7- WOC Ábaco .....	41
Gráfico 8- WOC CDStore .....	41
Gráfico 9- LOC Ábaco .....	42
Gráfico 10- LOC CDStore .....	42
Gráfico 11- DIT Ábaco .....	43
Gráfico 12- DIT CDStore .....	43
Gráfico 13- CDC Ábaco .....	44
Gráfico 14- CDC CDStore .....	44
Gráfico 15- CDO Ábaco .....	45
Gráfico 16- CDO CDStore .....	45
Gráfico 17- NOAConcern Ábaco .....	46
Gráfico 18- NOAConcern CDStore .....	46
Gráfico 19- NOOConcern Ábaco .....	47
Gráfico 20- NOOConcern CDStore .....	47
Gráfico 21- CDPDC Ábaco .....	48
Gráfico 22- CDPDC CDStore .....	48
Gráfico 23- CDPDO Ábaco .....	48
Gráfico 24- CDPDO CDStore .....	48
Gráfico 25- NOAConcernPDC Ábaco .....	49
Gráfico 26- NOAConcernPDC CDStore .....	49
Gráfico 27- CDTDC Ábaco .....	50
Gráfico 28- CDTDC CDStore .....	50
Gráfico 29- LOCCConcernPDC Ábaco .....	51
Gráfico 30- LOCCConcernPDC CDStore .....	51
Gráfico 31- TNOAConcern Ábaco .....	52

Gráfico 32- TNOAConcern CDStore .....	52
Gráfico 33- TLOCCConcern Ábaco .....	52
Gráfico 34- TLOCCConcern CDStore .....	52

## SUMÁRIO

<b>INTRODUÇÃO .....</b>	<b>10</b>
<b>CAPÍTULO 1 - CONCEITOS FUNDAMENTAIS .....</b>	<b>12</b>
1.1 - Considerações Iniciais .....	12
1.2 - Programação Orientada a Aspectos .....	12
1.3 - Métricas.....	15
<b>CAPÍTULO 2 - MÉTRICAS PARA COMPARAR SISTEMAS OO E OA .....</b>	<b>17</b>
2.1 - Considerações Iniciais .....	17
2.2 - Trabalhos Relacionados .....	17
2.3 - Sugestões De Novas Métricas e Regras Heurísticas.....	25
<b>CAPÍTULO 3 - ESTUDO DE CASO .....</b>	<b>28</b>
3.1 - Considerações Iniciais .....	28
3.2 - Sistemas Utilizados .....	28
3.3 - Aplicação das Métricas .....	29
3.3.1 - Métricas Tradicionais .....	29
3.3.2 - Métricas de Separação de Interesse .....	32
3.3.3 - Métricas Propostas .....	34
3.4 - Análise dos Resultados .....	37
3.4.1 - Métricas Tradicionais .....	37
3.4.2 - Métricas de Separação de Interesse .....	43
3.4.3 - Métricas Propostas .....	47
3.5 - Aplicação das Regras Heurísticas Propostas.....	53
<b>CONCLUSÃO .....</b>	<b>62</b>
<b>TRABALHOS FUTUROS .....</b>	<b>64</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>65</b>

Augusto, Rafael Henrique de Moraes  
Uma Comparação Entre Aplicações Orientadas A Objetos E  
Orientadas A Aspectos Com Uso De Métricas / Rafael Henrique de  
Moraes Augusto; orientador: Valter Vieira de Camargo.  
Marília, SP: [s.n], 2007.  
65 f.

Apresentação de Monografia (Graduação em Ciência da  
Computação) – Centro Universitário Eurípides de Marília –  
Fundação de Ensino Eurípides Soares da Rocha.

1. Métricas de Software. 2. DSOA. 3. Regras Heurísticas

CDD: 005-1

# INTRODUÇÃO

## Contexto

O Desenvolvimento de Software Orientado a Aspectos (DSOA) é um paradigma que visa a realizar uma separação avançada de interesses, com o objetivo de melhorar o nível de reusabilidade e manutenibilidade do software. Esse paradigma compreende novas abstrações da engenharia de software e trata diferentes dimensões de complexidade (Kiczales *et al.*, 1997). Conseqüentemente, alguns dos mecanismos de avaliação de *software* estão sendo adaptados para DSOA, como por exemplo, métricas de *software*.

Com os mecanismos de avaliação adaptados, é possível a realização de estudos comparativos que possam avaliar as vantagens e desvantagens do DSOA frente a outros paradigmas. Essa monografia busca apresentar uma comparação entre softwares orientados a aspectos (OA), e software orientados a objetos (OO). Para isso são utilizadas métricas adaptadas para o paradigma OA por Sant'anna (2004) e métricas elaboradas por Figueiredo (2006). Também estão sendo propostas novas métricas e regras heurísticas, para auxiliar a interpretação dos resultados obtidos.

## Motivação

Muitos pesquisadores já teceram conclusões a respeito de POA, porém, ainda não está bem claro as vantagens oferecidas por esse paradigma e o que o torna mais vantajoso em relação a paradigmas já consagrados como Orientação a Objeto. A motivação para esse trabalho é auxiliar o desenvolvedor de *software* a entender as vantagens de Orientação a Aspectos em relação à Orientação a Objeto e suas desvantagens em relação ao mesmo paradigma (Sant'anna, 2004, Figueiredo 2006).

## Objetivos

Nos últimos anos, alguns trabalhos foram desenvolvidos propondo a utilização de métricas de software para avaliar qualidades implícitas do Software como manutenibilidade e reusabilidade. Nesse trabalho são aplicadas algumas métricas em softwares com funcionalidades semelhantes, porém desenvolvidos em paradigmas diferentes (Paradigma

Orientado a Objetos e Paradigma Orientado a Aspectos). Com isso é possível averiguar quais são as vantagens e desvantagens de ambos os paradigmas.

### **Organização**

Essa monografia está dividida em três capítulos. No Capítulo 1, são apresentados os conceitos fundamentais para o entendimento do trabalho. No Capítulo 2, são apresentados trabalhos relacionados e métricas para comparação entre sistema Orientado a Objetos e sistemas Orientados a Aspectos. Ainda no Capítulo 2 serão propostas novas métricas. No Capítulo 3, é descrito um estudo de caso, utilizando sistemas já implementados nos dois paradigmas; neste sistema são aplicadas as métricas apresentadas no Capítulo 2 e avaliados os resultados obtidos com a aplicação das Métricas.

## CÁPITULO 1 - CONCEITOS FUNDAMENTAIS

### 1.1- Considerações Iniciais

Neste Capítulo são apresentados alguns conceitos fundamentais para o entendimento da Monografia. Na seção 1.2 são apresentados conceitos básicos de Programação Orientada a Aspectos (POA). Na seção 1.3 são apresentado um conceito sobre métricas para medir qualidades implícitas do software, como: reusabilidade e manutenibilidade.

### 1.2- Programação Orientada a Aspectos

A separação de interesses (*separation of concerns*) é um princípio bem estabelecido na engenharia de software que se refere à limitação da cognição humana em lidar com a complexidade de um software (Dijkstra, 1976). Dijkstra propõe uma estratégia que consiste em, a cada momento, focar a atenção em uma porção autocontida, ou interesse, isoladamente, tendo-se ciência de que se está ignorando temporariamente outras partes do problema.

Em 1997, Kiczales *et al.* (1997) publicaram um artigo apresentando a Programação Orientada a Aspectos. A sua idéia principal é que sistemas sejam programados de forma mais adequada distinguindo “interesses transversais” de “interesses-base”. Interesses-base referem-se à funcionalidade principal do sistema enquanto que um interesse transversal pode variar de requisitos de alto nível, como segurança e qualidade de um serviço, a noções de baixo nível, como sincronização e manipulações de *buffer* de memória. Podem ser tanto funcionais, como regras de negócio, quanto não-funcionais, como gerenciamento de transações e persistência de dados. Alguns exemplos mais comuns são: sincronização, interação de componentes, persistência, distribuição, gerenciamento de registros (*logging*) e controle de acesso (Kiczales *et al.*, 1997).

O termo “interesse transversal” faz analogia ao fato de que sua implementação com técnicas tradicionais de programação entrecorta transversalmente os módulos do sistema, afetando vários módulos e causando entrelaçamento (*tangling*) e espalhamento (*spreading*) de código de diferentes interesses. O entrelaçamento ocorre quando o código de um determinado

interesse encontra-se misturado com o código de um outro interesse dentro de um mesmo módulo. O espalhamento ocorre quando o código de um interesse encontra-se em vários módulos do sistema. O entrelaçamento e o espalhamento de código de diferentes interesses causam problemas de manutenção, reúso e evolução (Kiczales *et al.*, 1997).

A POA foi proposta como uma extensão da orientação a objetos, aproveitando todos os seus conceitos, porém adicionando conceitos específicos para a modularização dos interesses transversais como, por exemplo: aspectos (*aspects*) e adendos (*advices*). Assim os interesses transversais são implementados em módulos aspectuais e os interesses-base em classes normais. Uma das motivações para a criação da POA foi o reconhecimento de que os paradigmas de desenvolvimento de software convencionais, mais especificamente o orientado a objetos, sofrem da Tirania da Decomposição Dominante. Consiste na existência de apenas um tipo de módulo ou construção sintática – o módulo dominante – para se encapsular mais de um tipo de interesse (Tarr *et al.*, 2001). Alguns exemplos de módulos dominantes são as classes na programação orientada a objetos, as funções nas linguagens funcionais e as regras na programação baseada em regras. Um paradigma que permite a decomposição em apenas um tipo de módulo é chamado de unidimensional, pois há apenas uma dimensão para a modularização dos interesses, não importando quantos tipos de interesses existam.

A existência de apenas um tipo de módulo para encapsular mais de um tipo de interesse faz com que aqueles interesses que envolvem restrições globais – os interesses transversais – não fiquem adequadamente encapsulados e acabem misturando-se com os interesses-base.

A POA introduz novas abstrações de modularização e mecanismos de composição para melhorar a separação de interesses transversais. As novas abstrações propostas pela POA são: aspectos, pontos de junção (*join points*), conjuntos de junção (*pointcuts*), adendos e declarações intertipos (*inter-type declarations*).

Aspecto é o termo usado para denotar a abstração da POA que dá suporte a um melhor isolamento de interesses transversais. Em outras palavras, um aspecto corresponde a um interesse transversal e constitui uma unidade modular projetada para afetar um conjunto de classes e objetos do sistema.

Os pontos de junção são pontos bem definidos da execução de um sistema. Alguns exemplos de pontos de junção são: chamadas a métodos, execuções de métodos, leitura de atributos e modificação de atributos. Por meio dos pontos de junção, torna-se possível especificar o relacionamento entre aspectos e classes. Por exemplo, um aspecto A afeta uma classe C1 no instante da invocação de um método M1. Em outro exemplo, o aspecto A afeta

uma classe C2 ao final da execução de um método M2. O “instante da invocação do método M1” e o “ponto final da linha de execução do método M2” são exemplos de pontos de junção. Tais pontos de junção são os locais no programa onde os aspectos podem atuar. Eles podem ser especificados em uma linguagem orientada a aspectos por meio da definição de conjuntos de junção. Um conjunto de junção é o mecanismo que especifica os pontos de junção em que aspectos e classes se relacionam.

A implementação do comportamento transversal modularizado por um aspecto é feita em “adendos”. Adendo é um construtor semelhante a um método de uma classe, que define o comportamento dinâmico executado quando são alcançados um ou mais conjuntos de junção definidos previamente. Existem três tipos de adendos: os adendos anteriores (*before*), os posteriores (*after*) e os de substituição (*around*). Os adendos anteriores são executados sempre que os pontos de junção associados são alcançados e antes do prosseguimento da computação; os adendos posteriores são executados no término da computação, ou seja, depois que os pontos de junção forem executados e imediatamente antes do retorno do controle ao chamador; os adendos de substituição tomam o controle do código-base, podendo ou não devolver o controle.

As declarações intertipos podem ser utilizadas quando se deseja introduzir atributos e métodos em classes-base do sistema. Ao contrário de adendos, que operam de forma dinâmica, as declarações intertipos operam estaticamente, em tempo de compilação.

Um aspecto pode afetar a estrutura estática ou dinâmica de classes e objetos. A estrutura dinâmica é afetada por meio da especificação de conjuntos de junção e adendos; a estrutura estática é afetada por meio de declarações intertipo.

Além de especificarem elementos que entrecortam classes de um sistema, aspectos podem possuir métodos e atributos internos, como classes OO.

São três as propriedades básicas da POA (Kiczales *et al.*, 1997; Elrad *et al.*, 2001):

- dicotomia aspectos-base: diz respeito à adoção de uma distinção clara entre classes e aspectos. Os sistemas orientados a aspectos são decompostos em classes e aspectos. Os aspectos modularizam os interesses transversais e as classes modularizam os interesses-base;
- inconsciência: propriedade desejável da programação orientada a aspectos. É a idéia de que os componentes não precisam ser preparados para serem entrecortados por aspectos (Elrad *et al.*, 2001). Pela propriedade de inconsciência, os componentes não percebem os aspectos que poderão afetá-los. Embora seja uma propriedade interessante, algumas pesquisas atuais começam a apontar vantagens

quando o código-base possui consciência da existência dos aspectos (Kiczales e Mezini, 2005; Griswold, *et al.*, 2006);

- quantificação: capacidade de escrever declarações unitárias e separadas que afetam muitos pontos de um sistema (Elrad *et al.*, 2001). Pela propriedade de quantificação, é possível fazer declarações do tipo “em programas P, sempre que a condição C for verdadeira, faça a ação A”.

Combinador é o mecanismo responsável pela composição de classes e aspectos. Quase todo o processo de combinação é realizado como um pré-processamento em tempo de compilação. Entretanto, a versão 5 da linguagem AspectJ já oferece suporte ao processo de combinação dinâmico, pela associação de mecanismos da linguagem com frameworks como *AspectWerkz* (*AspectWerkz Website 2006*).

### 1.3 - Métricas

Medir a qualidade do *software* não é simples, consome muito tempo e esforço, em consequência do *software* ser um produto intangível. Pode se medir a qualidade pela taxa de erro do *software*, por exemplo quantidade de erros por milhões de linhas de código. A qualidade do *software* também pode ser definida como um conjunto de propriedades a serem satisfeitas em determinado grau, de modo que satisfaça as necessidades dos usuários e clientes. Essa definição aborda claramente qualidades explícitas do *software*, ou seja, características que podem ser documentadas na especificação de requisitos. Porém existe um conjunto de propriedades ou requisitos implícitos que devem ser frequentemente observados, como por exemplo o desejo de elevada flexibilidade, reusabilidade e manutenibilidade. Ou seja, o software que se adequar aos seus requisitos explícitos, mas deixar de cumprir os seus requisitos implícitos, sua qualidade será suspeita (Figueiredo, 2006).

Para alcançar um nível satisfatório de qualidade, os requisitos implícitos devem ser atendidos, sua medição não é trivial e requer medição por abordagens sistemáticas de avaliação. Para isso diversas métricas têm sido propostas na literatura, essas auxiliam a identificar onde é necessário efetuar melhorias nos artefatos gerados no processo de desenvolvimento.

As métricas consideradas mais relevantes são referentes ao código fonte porque esta é a forma mais confiável de informação. Métricas de código fonte devem ser fáceis de serem

obtidas, pois com diferentes versões de um software, há bastante informação a analisar (Figueiredo, 2005).

## **CÁPITULO 2 - MÉTRICAS PARA COMPARAR SISTEMAS OO E OA**

### **2.1- Considerações Iniciais**

Neste Capítulo são apresentadas Métricas, para comparação de Sistemas OO e OA. Na seção 2.2 são apresentados trabalhos relacionados onde foram utilizadas Métricas de software e regras heurísticas, para auxiliar na interpretação das Métricas. Na seção 2.3 são sugeridas novas métricas e novas regras heurísticas para interpretação.

### **2.2- Trabalhos Relacionados**

Sant'anna (2004) propôs um framework de avaliação para desenvolvimento de software orientado a aspectos (DSOA) composto por um conjunto de métricas e um modelo de qualidade. Como na época do trabalho não existiam métricas específicas para software orientado a aspectos, foram utilizadas algumas métricas clássicas, já existentes na literatura como, por exemplo, número de atributos e linhas de código, e também métricas orientadas a objetos que foram adaptadas para a orientação a aspectos. A adaptação realizada por Sant'anna (2004) consiste em denominar classes, interfaces e aspectos de “componentes” e os métodos e adendos de “operações”.

Os critérios utilizados por Sant'anna (2004) para escolher as métricas selecionadas foram: medir atributos de software (separação de interesse, acoplamento, coesão e tamanho); depender o máximo possível de métricas já consagradas e apontar as vantagens e desvantagens da orientação a aspectos comparada com a orientação a objeto.

A métrica “difusão do interesse em componentes (CDC)”, conta o número de componentes de um determinado interesse, que contribuem para a implementação do interesse avaliado. São contados também o número de componentes “secundários”, isto é, aqueles que acessam os componentes principais. Componentes principais são aqueles que implementam efetivamente o interesse, enquanto que “componentes secundários” são aqueles que usam o interesse implementado pelos principais. Essa métrica mede o grau de espalhamento do interesse pelos componentes. Quanto menos espalhado estiver o interesse, mais fácil será

entendê-lo para o reúso e menos componentes terão que ser alterados em um processo de manutenção.

Na métrica denominada “difusão do interesse em operações (CDO)”, deve-se contar o número de operações, construtores e métodos abstratos que contribuem para implementação do interesse avaliado. Também deve-se contar o número de operações “secundárias” que acessam qualquer operação principal. Com isso é possível medir o espalhamento do interesse em termos de operações. Quanto mais operações contribuírem para implementação do interesse, mais difícil será de entendê-lo, de reutilizá-lo e mais alterações serão necessárias em um processo de manutenção.

Outra métrica definida é a “difusão do interesse em linhas de código (CDLOC)”, essa tem o objetivo de avaliar quanto o código de um interesse está misturado com o código de outro interesse. Para isso é contado o número de “transições” entre o interesse avaliado e outros interesses. Por exemplo, o corpo de um método pode invocar três outros métodos de outras classes. Nesse caso o número de CDLOC seria seis, pois se deve contar a invocação de um método e o retorno ao interesse avaliado. Quanto maior o valor do CDLOC, mais misturado está o código do interesse. Quanto mais misturado o código do interesse, mais difícil de entendê-lo e mais difícil de reutilizá-lo.

A métrica “acoplamento entre componentes (CBC)” conta o número de classes que são usadas como tipos de atributos, parâmetro de operações, tipos de retorno ou variáveis locais. Conta-se também para cada aspecto, as classes nas quais são introduzidos atributos e métodos por meio de declaração inter-tipo, as classes ou aspectos que são interceptados pelo aspecto por meio de definições de conjunto de junção e os componentes cujo atributos e métodos introduzidos via declaração inter-tipo são acessados pelo aspecto. Essa métrica foi definida por que a compreensão de um componente envolve o entendimento dos componentes acoplados. Dessa forma quanto maior o número de acoplamentos mais difícil de entendê-lo e maior será a sensibilidade a mudanças em outras partes do projeto e assim tornar a manutenção mais difícil.

A “Profundidade da árvore de herança (DIT)”, é uma métrica que mede o comprimento máximo de um nó até a raiz da árvore. Para contar o valor de DIT deve-se avaliar os componente. Um componente que não estende nenhum outro componente tem valor de DIT igual a um (1). Um componente que estende outro componente tem valor de DIT igual a dois (2), e assim por diante. Essa métrica mede a profundidade da herança que um componente é declarado. Quanto mais profundo na hierarquia um componente for declarado mais difícil será de entendê-lo e reutilizá-lo.

A “Falta de coesão em operações (LCOO)”, é uma métrica que mede a falta de coesão de um componente. A LCOO mede a quantidade de pares de métodos / adendo, que não acessam um mesmo atributo. Também são considerados os métodos que o aspecto introduz nas classes que ele afeta. Por exemplo, para calcular o valor de LCOO para um componente, deve-se começar pela sua primeira operação e verificar se ele manipula atributos em comum com cada operação abaixo dela. Após isso, deve-se verificar se a segunda operação manipula atributos em comum com cada operação abaixo dela, a assim por diante. Componentes com baixa coesão sugerem um projeto inadequado, pois não estão bem encapsulados.

A métrica “tamanho do vocabulário (VS)” conta o número de componentes do sistema. As instanciações não são contadas. Quanto maior o VS do sistema mais difícil de entender e reutilizar.

Linha de código (LOC) é uma métrica que visa a contar o número de linhas de código de forma tradicional. Comentários e linhas em branco não são contados. Quanto maior o número de linhas de código, maior a dificuldade de entender o sistema e maior a dificuldade de localizar o ponto que deve ser alterado em um processo de manutenção e evolução.

Número de atributos (NOA) mede o número de atributos de cada classe ou aspecto. Atributos herdados não são contados por essa métrica. No aspecto também não devem ser contados os atributos que ele introduz nas classes afetadas. Quanto maior o número de atributos, mais difícil de entender o sistema e mais difícil de encontrar os locais que devem ser alterados em um processo de manutenção.

Número de operações (NOO) mede o número de operações de cada classe ou aspecto. Quanto maior o número de operações, mais difícil de entender o sistema e mais difícil de encontrar os locais que devem ser alterados em um processo de manutenção.

Peso de operação por componente (WOC), essa métrica mede a complexidade de um componente em termos de suas operações. A complexidade é computada contando-se o número de parâmetros das operações, uma operação sem parâmetros tem complexidade 1 (um), uma operação com um parâmetro tem complexidade 2 (dois) e assim por diante. Quanto maior a complexidade das operações por componentes, mais difícil de entender o sistema e mais difícil de realizar processos de manutenção.

Com o intuito de complementar o estudo realizado por Sant’anna (2004), Figueiredo (2006), propõe uma abordagem que provê suporte à avaliação quantitativa para o desenvolvimento de software orientado a aspectos. Nessa abordagem foi proposto um método de avaliação organizado em etapas e uma ferramenta, denominada AJATO, para automatizar

o método. Ao método proposto por Figueiredo (2006), inclui-se ainda o conjunto de métricas proposto por Sant'anna (2004), e três (3) novas métricas e regras heurísticas, propostas pelo próprio Figueiredo (2006).

A métrica “número de atributos do interesse (NOAconcern)” conta o número de atributos de um componente cujo propósito principal é a implementação do interesse avaliado. Essa métrica mede o grau de espalhamento de um interesse pelos atributos de um componente. Mede também quanto os atributos são destinados à implementação do interesse avaliado. Quanto menos atributos destinados ao interesse, menor a dedicação do componente àquele interesse, o que indica que esse interesse provavelmente deve ser modularizado em outro componente.

A métrica “número de operações de interesse (NOOconcern)” conta o número de operações de um componente cujo propósito principal é implementar o interesse avaliado. Essa mede o grau de espalhamento do interesse pelas operações de um componente. Quanto menos elementos do componente são destinados ao interesse, menor é a dedicação do componente ao interesse. Um componente que possui poucas operações cujo propósito é implementar o interesse, provavelmente, indica que o interesse deve ser separado em outro componente.

A métrica “número de linhas de código do interesse (LOCconcern)” conta o número de linhas de código de um componente cujo propósito principal é implementar o interesse avaliado. Essa mede o grau de espalhamento de um interesse pelos elementos do componente, se poucos elementos do componente são destinados ao interesse avaliado, provavelmente esse interesse seja melhor separado utilizando técnicas DSOA.

Além do conjunto de métricas, o método de avaliação proposto por Figueiredo (2006) é composto por um conjunto de regras heurísticas que são aplicadas sobre o resultado das medições. O objetivo dessas regras é apontar potenciais problemas no projeto ou implementação relacionados aos interesses transversais e que não são trivialmente detectáveis. Estes problemas incluem: interesses que não são facilmente identificados como interesses transversais; partes de um interesse transversal que não foi modularizado em um projeto OA; interesses entrelaçados e espalhados; aspectos com problemas de coesão. As regras heurísticas contribuem para um melhor entendimento dos números encontrados pelas métricas. Na Tabela 1 é apresentado um conjunto de regras heurísticas derivadas de uma única métrica. Por exemplo, aplicando-se a métrica CDC (Sant'anna, 2004) é possível avaliar se um interesse afeta vários componentes do sistema, isto é, se o interesse está espalhado, e sugerir ao desenvolvedor refatorações no código. Refatorações são alterações feitas nos artefatos de

software que não alteram o comportamento observável do sistema. O uso desse mecanismo tem como objetivo deixar o código mais reutilizável, manutenível e fácil de ser entendido.

**Tabela 1– Conjunto de regras heurísticas base em uma única métrica - Figueiredo (2006)**

Regras Heurísticas		Métricas
01	Se um interesse afeta vários componentes do sistema Então este é um interesse espalhado.	CDC
02	Se o código de um interesse se mistura ao código de outros interesses Então esse é um interesse entrelaçado.	CDLOC
03	Se os atributos e operações de um componente não são fortemente relacionado Então esse é um componente pouco coeso	LCOO
04	Se um componente depende muito de outros componentes Então esse é um componente altamente acoplado	CBC
05	Se um componente está muito distante da raiz na hierarquia de herança Então esse é um componente de herança profunda.	DIT

As duas primeiras regras da Tabela 2 podem ser usadas para alertar o desenvolvedor da existência de um interesse, possivelmente transversal, que se encontra espalhado e entrelaçado no código. Essas duas regras são suportadas respectivamente, pelas métricas CDC e CDLOC. Nessa tabela também são apresentadas outras três regras que são utilizadas para identificar componentes com baixa coesão (Regra 03) ou alto acoplamento (Regras 04 e 05). As três última regras são suportadas pelas métricas LCOO, CBC e DIT.

Na Tabela 2 é apresentado um conjunto de regras heurísticas que utilizam uma combinação de métricas de separação de interesses e tamanho, para avaliar os interesses do sistema e identificar possíveis problemas relacionados a interesses transversais. As métricas utilizadas são CDC, CDLOC, NOAConcern, NOOConcern, LOCCConcern, VS, NOA, NOO e LOC. O objetivo principal do conjunto de regras é classificar o interesse avaliado em “Modularizado”, “Primário” ou “Secundário”. As duas primeiras classificações de interesses não indicam problemas de separação de interesse, enquanto a última pode ser problemática e, portanto, indicar que o projeto deve ser refatorado para melhor modularizar esse tipo de interesse. Além das três categorias as regras também classificam o interesse como entrelaçado, de baixo espalhamento e de elevado espalhamento. Interesse modularizado é aquele onde todos os componentes responsáveis pela sua implementação são totalmente

dedicados a este interesse. Em cada componente do sistema é definido um conjunto de dados e funções que, idealmente, atendem a um mesmo propósito, este é o que chamamos de interesse primário do componente, porém um interesse só será classificado como primário se ele é primário em todos os componentes em que ele se encontra. Um interesse é classificado como entrelaçado quando este se encontra misturado a outros interesses dentro dos componentes. Um componente dedicado ao interesse primário, que implemente outros interesses é classificado como Interesse secundário. O espalhamento é a característica do interesse relacionada ao número de componentes afetados por ele assim, quando vários componentes implementam o interesse ele é classificado como elevado espalhamento e quando apenas alguns componentes implementam o interesse ele é classificado como baixo espalhamento.

**Tabela 2 – Conjunto de regras heurísticas de separação de interesse – Figueiredo (2006)**

Regras Heurísticas de separação de interesse	
R01	Se CDLOC é 2 Então interesse modularizado
R02	Se CDLOC é maior que 2 Então interesse entrelaçado
R03	Se CDC / VS de interesse entrelaçado é alto Então interesse de elevado espalhamento
R04	Se CDC / VS de interesse entrelaçado não é alto Então interesse de baixo espalhamento
R05	Se (NOACConcern / NOA é baixo) e (NooConcern / NOO é baixo) para todos os componentes de interesse de elevado espalhamento Então interesse possivelmente primário
R06	Se (NOACConcern / NOA é baixo) e (NooConcern / NOO é baixo) para pelo menos um componente de interesse de elevado espalhamento Então interesse possivelmente secundário
R07	Se (NOACConcern / NOA é baixo) e (NooConcern / NOO é baixo) para todos os componentes de interesse de abaixo espalhamento Então interesse possivelmente primário
R08	Se (NOACConcern / NOA é baixo) e (NooConcern / NOO é baixo) para pelo menos um componente de interesse de baixo espalhamento Então interesse possivelmente secundário
R09	Se (LOCCConcern / LOC é baixo) para todos os componentes de possível interesse primário Então interesse primário.
R10	Se (LOCCConcern / LOC é baixo) para pelo menos um componente de possível interesse secundário Então interesse secundário.

As regras que compõem o grupo mostrado na Tabela 2 devem ser aplicadas sequencialmente, na ordem que elas são numeradas, de tal forma que a classificação do interesse é gradativamente refinada. As categorias não são exclusivas, ou seja, a nova classificação dada por uma regra é acumulativa com a dada pela regra anterior.

As duas primeiras regras da Tabela 2 (R01 e R02) utilizam a métrica CDLOC para classificar o interesse em “Modularizado” ou “Entrelaçado”. As regras R03 e R04 são usadas para verificar se o interesse, quando classificado como entrelaçado, se espalha por um grande número de componentes ou não. Tanto um “Interesse de Elevado Espalhamento”, quanto um “Interesse de Baixo Espalhamento”, pode ser primário ou secundário. As regras R05 e R06 são usadas para decidir se um “Interesse de Elevado Espalhamento” é “Possivelmente Primário” ou “Possivelmente Secundário”, enquanto as regras R07 e R08 fazem análise semelhante para um “Interesse de Baixo Espalhamento”. A regra R09 classifica o interesse como “Primário” se este é “Possivelmente Primário” e a maior parte das linhas de código é destinada a implementação deste interesse. A regra R10 classifica o interesse como “Secundário” se este é “Possivelmente Secundário” e a maior parte das linhas de código não se destinada a implementação deste interesse.

A interpretação das regras depende de valores e limites que quando ultrapassados, disparam as conseqüências das regras. O valores limites encontrados para as regras por Figueiredo (2006) é apresentado na Tabela 2.1.

**Tabela 2.1– Valores limites utilizados nas regras heurísticas – Figueiredo (2006)**

Regras	Valores Limites
Regra 01 e 02	2
Regras 03 e 04	50% (se $VS \leq 10$ ) 40% (se $10 < VS \leq 20$ ) 30% (se $20 < VS \leq 50$ ) 20% (se $50 < VS \leq 80$ ) 15% (se $80 < VS$ )
Regras 05, 06, 07 e 08	50% (de NOA) 50% (de NOO)
Regras 09 e 10	50% (de LOC)

Com o intuito de automatizar o processo de aplicação das métricas e regras heurísticas Figueiredo (2006) propôs uma ferramenta denominada AJATO, que deve ser usada na fase de implementação. Essa é composta por quatro módulos que efetuam o *parser* do código, mapeamento da estrutura sintática em interesse, medição e avaliação heurística.

Essa ferramenta automatiza a aplicação das métricas e sugere por meio das regras heurísticas alterações no projeto. No entanto, isso não garante que o software precisa ser alterado. Figueiredo (2006) afirma que as alterações propostas pela ferramenta, são advertências e não dispensam a avaliação de um desenvolvedor, pois em alguns casos essa alterações não são cabíveis.

Para que as métricas sejam computadas pela ferramenta é necessário, a classificação do interesse de cada operação e cada componente do sistema que será avaliado. A Figura 1 mostra o componente “OORelationalMapping”, do projeto Ábaco-OA que é descrito na Seção 3.2 sendo classificado no interesse de persistência.

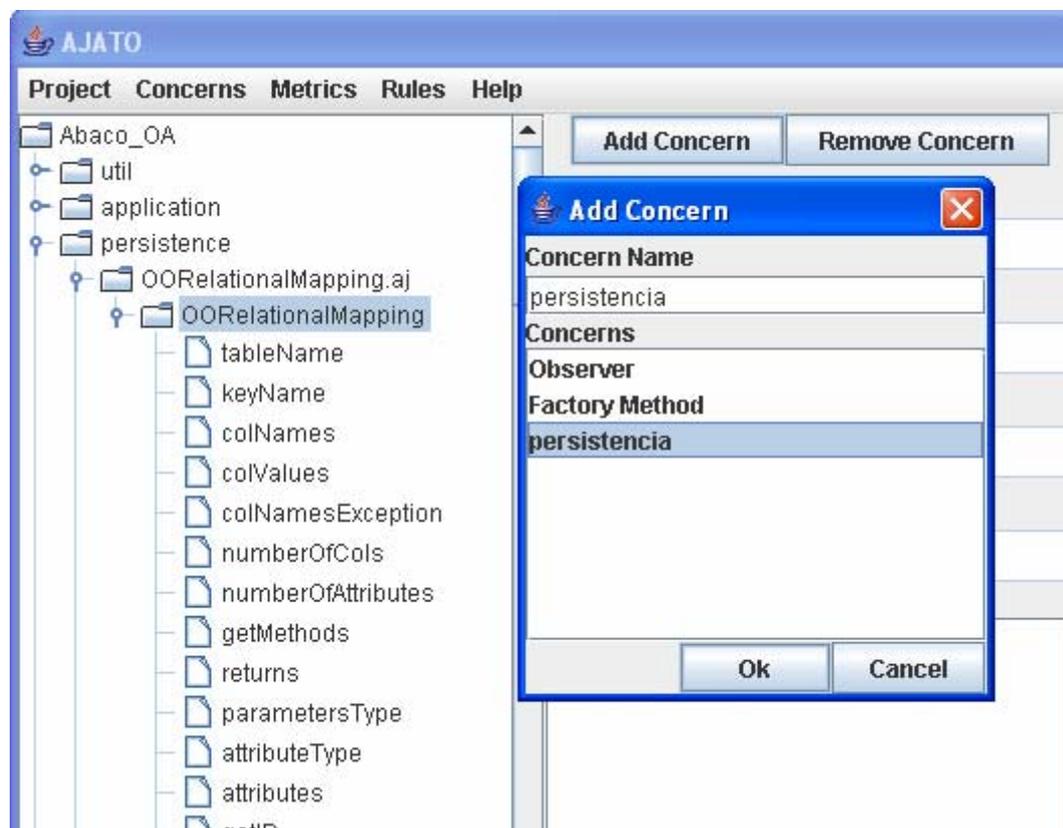


Figura 1 – Interface da ferramenta AJATO – Figueiredo (2006)

Ramos *et al.* (2006) propõem uma abordagem de instanciação de métricas para medir documentos de requisitos orientados a aspectos, buscando alternativas para obtenção de documentos de requisitos mais estruturados. Essa abordagem propõe tornar as métricas CDC, VS, NOE, NOCE e CBC, abstratas para que possam ser utilizadas (quanto instanciadas) em um documento de requisito orientado a aspectos. Embora Ramos *et al.* (2006) apresente alguma pesquisa relacionada a métricas, seu trabalho não tem forte relacionamento com este.

### 2.3 - Sugestão de Novas Métricas e Regras Heurísticas

As métricas de separação de interesse propostas nos trabalhos de Figueiredo (2006) e Sant'ana (2004), auxiliam na identificação de um interesse que é possivelmente irá se tornar um interesse transversal. Porém quando utilizadas para comparação de aplicações semelhantes desenvolvidas, sendo uma desenvolvida em OA e outra em OO, apenas com essas regras é difícil avaliar a qualidade das aplicações quando comparadas uma com a outra. Para auxiliar nesta avaliação neste trabalho são propostas algumas métricas e regras heurísticas. Essas métricas são descritas abaixo.

A primeira métrica proposta é “Difusão do interesse em operações parcialmente dedicadas” (“*Concern Difusion in Partially Dedicated Operations*” - CDPDO), conta o número de operações parcialmente dedicadas à implementação do interesse. Operações parcialmente dedicadas são operações que contêm linhas de código que contribuem para a implementação do interesse avaliado e contêm linhas de código que contribuem para implementação de outro interesse. CDPDO é uma métrica complementar a métrica NOOConcern, proposta por Figueiredo (2006), a soma do valor das duas métricas deve ser igual ao resultado da métrica CDO. A métrica CDPDO mede o quão entrelaçado está o interesse nas operações dos componentes parcialmente dedicados. Quanto maior o número de operações em componentes parcialmente dedicados, mais espalhado está o interesse.

Difusão do interesse em componentes parcialmente dedicados (“*Concern Difusion in Partially Dedicated Component*” - CDPDC), conta o número de componentes parcialmente dedicados à implementação do interesse avaliado. Um componente parcialmente dedicado a implementação de um interesse é aquele que contém pelo menos uma operação totalmente ou parcialmente dedicada ao interesse avaliado e pelo menos uma operação que envolve algum outro interesse. Um componente parcialmente dedicado é aquele que o CDO é maior que zero e menor que o total de operações. Essa métrica mede o quão espalhado está o interesse nos componentes parcialmente dedicados. Quanto maior o número de componentes parcialmente dedicados em relação ao número total de componentes mais espalhado está o interesse.

Difusão do interesse em componentes totalmente dedicados (“*Concern Difusion in Total Dedicated Component*” - CDTDC), conta o número dos componentes totalmente dedicados à implementação do interesse avaliado. Um componente totalmente dedicado à

implementação de um interesse é aquele que contém apenas operações totalmente dedicadas ao interesse avaliado. Essa métrica serve de apoio para a interpretação dos resultados.

“Número de atributos do interesse em componentes parcialmente dedicados” (“Number of Attributes *in Partially Dedicated Component*” - NOAConcernPDC), conta o número de atributos dedicados à implementação do interesse, dos componentes parcialmente dedicados ao interesse avaliado. Essa métrica mede o quão espalhado está o interesse nos atributos dos componentes parcialmente dedicados. Quanto maior o número de atributos em componentes parcialmente dedicados mais espalhado está o interesse.

“Número total de atributos do interesse” (“Total Number of Attributes *Concern*” - TNOAConcern), conta o número total de atributos dedicados à implementação do interesse, de todos os componentes dedicados ao interesse avaliado. Ou seja esta métrica é a soma dos valores encontrados nas métricas NOAConcern e NOAConcernPDC. Essa métrica serve de apoio para a aplicação das regras heurísticas que serão propostas nesse trabalho.

“Número de linhas de código do interesse em componentes parcialmente dedicados” (“Lines of Code per Concern *in Partially Dedicated Component*” - LOCCConcernPDC), conta o número de linhas de código dedicadas à implementação do interesse, dos componentes parcialmente dedicados ao interesse avaliado. Essa métrica mede o quão espalhado está o interesse em linhas de código dos componentes parcialmente dedicados. Quanto maior o número de linhas de código em componentes parcialmente dedicados mais entrelaçado está o interesse.

“Número total de linhas de código do interesse” (“Total Line of Code per Concern” - TLOCCConcern), conta o número total de linhas de código dedicadas à implementação do interesse, ou seja esta métrica é a soma dos valores encontrados pelas métricas LOCCConcern e TLOCCConcernPDC. Essa métrica serve de apoio para a aplicação das regras heurísticas que serão propostas nesse trabalho.

Além das métricas sugeridas nessa Seção é apresentado um conjunto de sugestões de regras heurísticas que apóia na comparação de aplicações semelhantes desenvolvidas, sendo uma desenvolvida em OA e outra e OO. Estas são apresentadas na Tabela 3.

**Tabela 3– Conjunto de regras heurísticas sugeridas**

Regras Heurísticas Comparativas	
R01	<p>Quanto maior o percentual de CDPDO sobre CDO, mais entrelaçados estão as operações do interesse.</p> <p>Exemplo: Se o valor da métrica CDPDO é igual a 50 e valor da métrica CDO é igual a 100, isso indica que as operações do interesse avaliado estão 50 % entrelaçadas no código base.</p>
R02	<p>Quanto maior o percentual de CDPDO sobre NOOConcern, mais entrelaçados estão as operações do interesse.</p> <p>Exemplo: Se o valor da métrica CDPDO é igual a 30 e valor da métrica NOOConcern é igual a 100, isso indica que as operações parcialmente dedicadas a implementação interesse avaliado representam 30 % das operações totalmente dedicadas ao interesse e esses 30 % estão entrelaçadas pelo código base.</p>
R03	<p>Quanto maior o percentual de CDPDC sobre CDC, mais espalhados estão os componentes do interesse.</p> <p>Exemplo: Se o valor da métrica CDPDC é igual a 5 e valor da métrica CDC é igual a 10, isso indica que os componentes do interesse avaliado estão 50 % espalhados no código base.</p>
R04	<p>Quanto maior o percentual de CDPDC sobre CDTDC, mais espalhados estão os componentes do interesse.</p> <p>Exemplo: Se o valor da métrica CDPDC é igual a 3 e valor da métrica CDTDC é igual a 10, isso indica que os componentes parcialmente dedicados a implementação interesse avaliado representam 30 % dos componentes totalmente dedicados e esses 30% estão espalhados no código base.</p>
R05	<p>Quanto maior o percentual de NOAConcernPDC sobre TNOAConcern mais espalhado estão os atributos do interesse.</p> <p>Exemplo: Se o valor da métrica NOAConcernPDC é igual a 50 e valor da métrica TNOAConcern é igual a 100, isso indica que os atributos do interesse avaliado estão 50 % espalhados no código base.</p>
R06	<p>Quanto maior o percentual de NOAConcernPDC sobre NOAConcern, mais espalhado estão os atributos do interesse.</p> <p>Exemplo: Se o valor da métrica NOAConcernPDC é igual a 30 e valor da métrica NOAConcern é igual a 100, isso indica que os atributos parcialmente dedicados ao interesse avaliado representam 30 % dos atributos totalmente dedicados ao interesse e esses estão espalhados no código base.</p>
R07	<p>Quanto maior o percentual de LOCCConcernPDC sobre TLOCCConcern, mais entrelaçado estão as operações do interesse.</p> <p>Exemplo: Se o valor da métrica LOCCConcernPDC é igual a 500 e valor da métrica TLOCCConcern é igual a 1000, isso indica que as linhas de código do interesse avaliado estão 50 % entrelaçadas no código base.</p>
R08	<p>Quanto maior o percentual de LOCCConcernPDC sobre LOCCConcern, mais entrelaçado estão as operações do interesse.</p> <p>Exemplo: Se o valor da métrica LOCCConcernPDC é igual a 300 e valor da métrica LOCCConcern é igual a 1000, isso indica que as linhas de código de componentes parcialmente dedicados ao interesse avaliado representam 30 % das linhas de código de componentes totalmente dedicados ao interesse e essas estão entrelaçadas no código base.</p>

## CÁPITULO 3 – ESTUDO DE CASO

### 3.1 - Considerações Iniciais

Neste Capítulo é apresentado um estudo de caso, aplicando as métricas apresentadas em sistemas OA e OO com as mesmas funcionalidades. Na seção 3.2, são apresentados os sistemas utilizados. Na seção 3.3, são apresentados os resultados obtidos com as aplicações das métricas. Na seção 3.4, é feita uma análise dos resultados obtidos com a aplicação das métricas. Na seção 3.5, é apresentada a aplicação das regras heurísticas propostas. Para uma melhor diferenciação das métricas de separação de interesse, das métricas (clássicas e orientadas a objetos, adaptadas por Sant’anna - 2004) serão chamadas de métricas tradicionais.

### 3.2- Sistemas Utilizados

Neste estudo de caso foram utilizados os *softwares* Ábaco e CDStore. Estes foram desenvolvidos por Camargo (2006) ambos foram desenvolvidos nas versões OA e OO, com as mesmas funcionalidades e implementando o interesse de persistência.

CDStore é um sistema para loja *CDs*, trata do gerenciamento das vendas de CDs a clientes cadastrados. Cada CD possui uma etiqueta com uma cor que determina o seu preço, o qual pode ser reajustado a qualquer momento pelo gerente da loja. Diversos relatórios devem ser gerados pelo sistema para permitir a gestão adequada da loja, como o relatório de clientes que compraram CD em uma determinada data/período, relatório dos CD mais vendidos em um determinado período, relatório dos clientes mais assíduos, etc.

O Ábaco é um sistema para Gestão de Oficina de Eletro-eletrônicos (“SIGOE”) tem por objetivo principal a gestão dos consertos efetuados em aparelhos elétricos ou eletrônicos e a manutenção do estoque de peças. A oficina possui uma página na Internet que possibilita alguns serviços ao cliente, tais como: cadastro de dados pessoais, requisição de um conserto em seu aparelho, consulta à situação do conserto atual, consulta a contas a pagar, consulta ao histórico de consertos do seu aparelho, etc. Existe também o serviço de coleta e entrega do aparelho, tanto para requisições pela Internet como por telefone. A oficina possui

*funcionários* com diferentes especialidades. A realização do conserto pode utilizar peças em estoque, que devem ser debitadas automaticamente do estoque. Os *funcionários* recebem percentuais de comissão sobre o serviço de mão-de-obra executado, conforme acordo com o proprietário da oficina. O sistema deve também permitir o controle de compras efetuadas dos *fornecedores*, que podem ser precedidas de pedidos ou podem ser feitas diretamente.

Para medir as métricas foi utilizado como ferramenta de apoio o software AJATO desenvolvido por Figueiredo (2006), que foi descrito no Capítulo 2.

### 3.3 – Aplicação das Métricas

As métricas apresentada na Seção 1.2, foram aplicadas nos sistemas apresentados na Seção 2.1. Na Seção 2.2.1 serão demonstrados os resultados obtidos com a aplicação das métricas tradicionais OO, adaptadas para OA por Sant’anna (2004). Na Seção 2.2.2 serão demonstrados os resultados obtidos com a aplicação das métricas de separação de interesse, propostas por Sant’anna (2004) e Figueiredo (2006). Na Seção 2.2.3 serão demonstrados os resultados obtidos com a aplicação das métricas sugeridas neste trabalho.

#### 3.3.1 – Métricas Tradicionais

Na Tabela 4 são apresentados os resultados obtidos com a aplicação da métrica VS (tamanho do vocabulário). No software ábaco VS é igual a 57 (cinquenta e sete) na versão OA e 49 (quarenta e nove) na versão OO, no software CDStore os resultados são 46 (quarenta e seis) para a versão OA e 40 (quarenta) para a versão OO.

**Tabela 4 – Aplicação da métrica VS**

	VS
Ábaco-OA	57
Ábaco-OO	49
CDStore-OA	46
CDStore-OO	40

Na Tabela 5 são apresentados os resultados obtidos com a aplicação da métrica NOA (Número de atributos). No software ábaco NOA resulta em 80 (oitenta) na versão OA e 79 (setenta e nove) na versão OO. No software CDStore os resultados são 51 (cinquenta e um) na versão OA e 50 (cinquenta) para a versão OO.

**Tabela 5 – Aplicação da métrica NOA**

	NOA
Ábaco-OA	80
Ábaco-OO	79
CDStore-OA	51
CDStore-OO	50

Na Tabela 6 são apresentados os resultados obtidos com a aplicação da métrica NOO (Número de operações). No software ábaco NOO resulta em 329 (trezentos e vinte e nove) na versão OA e 320 (trezentos e vinte) na versão OO. No software CDStore os resultados são 230 (duzentos e trinta e sete) na versão OA e 229 (duzentos e vinte e nove) para a versão OO.

**Tabela 6 – Aplicação da métrica NOO**

	NOO
Ábaco-OA	329
Ábaco-OO	320
CDStore-OA	237
CDStore-OO	229

Na Tabela 7 são apresentados os resultados obtidos com a aplicação da métrica WOC (Peso de operação por componente). No software ábaco WOC resulta em 604 (seiscentos e quatro) e na versão OA e 590 (quinhentos e noventa) na versão OO. No software CDStore os resultados são 438 (quatrocentos e trinta e oito) na versão OA e 425 (quatrocentos e vinte e cinco) para a versão OO.

**Tabela 7 – Aplicação da métrica WOC**

	WOC
Ábaco-OA	604
Ábaco-OO	590
CDStore-OA	438
CDStore-OO	425

Na Tabela 8 são apresentados os resultados obtidos com a aplicação da métrica LOC (Linhas de código). No software ábaco LOC resulta em 4093 (quatro mil e noventa e três) na versão OA e 3930 (três mil novecentos e trinta) na versão OO. No software CDStore os resultados são 3529 (três mil quinhentos e vinte e nove) de na versão OA e 3396 (três mil trezentos e noventa e seis) para a versão OO.

**Tabela 8 – Aplicação da métrica LOC**

	LOC
Ábaco-OA	4093
Ábaco-OO	3930
CDStore-OA	3529
CDStore-OO	3396

Na Tabela 9 são apresentados os resultados obtidos com a aplicação da métrica DIT (Profundidade da árvore de herança). No software ábaco DIT resulta em 83 (oitenta e três) na versão OA e 85 (oitenta e cinco) na versão OO. No software CDStore os resultados foram 65 (sessenta e cinco) na versão OA e 64 (sessenta e quatro) para a versão OO.

**Tabela 9 – Aplicação da métrica DIT**

	DIT
Ábaco-OA	83
Ábaco-OO	85
CDStore-OA	65
CDStore-OO	64

### 3.3.2 – Métricas de Separação de Interesse

Na Tabela 10 são apresentados os resultados obtidos com a aplicação da métrica CDC (Difusão do interesse em componentes). No software ábaco CDC resulta em 22 (vinte e dois) na versão OA e 33 (trinta e três) na versão OO. No software CDStore os resultados foram 20 (vinte) na versão OA e 31 (trinta e um) para a versão OO.

**Tabela 10 – Aplicação da métrica CDC**

	CDC
Ábaco-OA	22
Ábaco-OO	33
CDStore-OA	20
CDStore-OO	31

Na Tabela 11 são apresentados os resultados obtidos com a aplicação da métrica CDO (Difusão do interesse em operações). No software ábaco CDO resulta em 147 (cento e quarenta e sete) na versão OA e 152 (cento e cinquenta e dois) na versão OO. No software CDStore os resultados foram 147 (cento e quarenta e sete) na versão OA e 156 (cento e cinquenta e seis) para a versão OO.

**Tabela 11 – Aplicação da métrica CDO**

	CDO
Ábaco-OA	147
Ábaco-OO	152
CDStore-OA	147
CDStore-OO	156

Na Tabela 12 são apresentados os resultados obtidos com a aplicação da métrica NOAConcern (Número de atributos de interesse). No software ábaco NOAConcern resulta em 27 (vinte e sete) na versão OA e 26 (vinte e seis) na versão OO. No software CDStore os

resultados foram os mesmos 27 (vinte e sete) na versão OA e 26 (vinte e seis) para a versão OO.

**Tabela 12 – Aplicação da métrica NOAConcern**

	NOAConcern
Ábaco-OA	27
Ábaco-OO	26
CDStore-OA	27
CDStore-OO	26

Na Tabela 13 são apresentados os resultados obtidos com a aplicação da métrica NOOConcern (Número de operações de interesse). No software ábaco NOOConcern resulta em 147 (cento e quarenta e sete) na versão OA e 133 (cento e trinta e três) na versão OO. No software CDStore os resultados foram 147 (cento e quarenta e sete) na versão OA e 139 (cento trinta e nove) para a versão OO.

**Tabela 13 – Aplicação da métrica NOOConcern**

	NOOConcern
Ábaco-OA	147
Ábaco-OO	133
CDStore-OA	147
CDStore-OO	139

Na Tabela 14 são apresentados os resultados obtidos com a aplicação da métrica LOCCConcern (Linhas de código do interesse). No software ábaco LOCCConcern resulta em 2409 (dois mil quatrocentos e nove) na versão OA e 2273 (dois mil duzentos e setenta e três) na versão OO. No software CDStore os resultados foram 2409 (dois mil quatrocentos e nove) na versão OA e 2268 (dois mil duzentos e sessenta e oito) para a versão OO.

**Tabela 14 – Aplicação da métrica LOCCConcern**

	LOCCConcern
Ábaco-OA	2409
Ábaco-OO	2273
CDStore-OA	2409
CDStore-OO	2268

### 3.3.3 – Métricas Propostas

Na Tabela 15 são apresentados os resultados obtidos com a aplicação da métrica CDPDC (difusão do interesse em componentes parcialmente dedicados). No software ábaco CDPDC resulta em 0 (zero) na versão OA e 20 (vinte) na versão OO. No software CDStore os resultados foram 0 (zero) na versão OA e 17 (dezesete) para a versão OO.

**Tabela 15 – Aplicação da métrica CDPDC**

	CDPDC
Ábaco-OA	0
Ábaco-OO	20
CDStore-OA	0
CDStore-OO	17

Na Tabela 16 são apresentados os resultados obtidos com a aplicação da métrica CDPDO (difusão do interesse em operações parcialmente dedicadas). No software ábaco CDPDO resulta em 0 (zero) na versão OA e 19 (dezenove) na versão OO. No software CDStore os resultados foram 0 (zero) na versão OA e 17 (dezesete) para a versão OO.

**Tabela 16 – Aplicação da métrica CDPDO**

	CDPDO
Ábaco-OA	0
Ábaco-OO	19
CDStore-OA	0
CDStore-OO	17

Na Tabela 17 são apresentados os resultados obtidos com a aplicação da métrica NOAConcernPDC (número de atributos do interesse em componentes parcialmente dedicados). No software ábaco NOAConcernPDC resulta em 0 (zero) na versão OA e 11 (onze) na versão OO, no software CDStore os resultados foram 0 (zero) na versão OA e 12 (doze) para a versão OO.

**Tabela 17 – Aplicação da métrica NOAConcernPDC**

	NOAConcernPDC
Ábaco-OA	0
Ábaco-OO	11
CDStore-OA	0
CDStore-OO	12

Na Tabela 18 são apresentados os resultados obtidos com a aplicação da métrica CDTDC (difusão do interesse em componentes totalmente dedicados). No software ábaco TD-CDC resulta em 22 (vinte e dois) na versão OA e 13 (treze) na versão OO, no software CDStore os resultados foram 20 (vinte) na versão OA e 14 (quatorze) para a versão OO.

**Tabela 18 – Aplicação da métrica CDTDC**

	CDTDC
Ábaco-OA	22
Ábaco-OO	13
CDStore-OA	20
CDStore-OO	14

Na Tabela 19 são apresentados os resultados obtidos com a aplicação da métrica LOCConcernPDC (número de linhas de código do interesse em componentes parcialmente dedicados). No software ábaco LOCConcernPDC resulta em 0 (zero) na versão OA e 57 (cinquenta e sete) na versão OO, no software CDStore os resultados foram 0 (zero) na versão OA e 51 (cinquenta e um) para a versão OO.

**Tabela 19 – Aplicação da métrica LOCConcernPDC**

	LOCConcernPDC
Ábaco-OA	0
Ábaco-OO	57
CDStore-OA	0
CDStore-OO	51

Na Tabela 20 são apresentados os resultados obtidos com a aplicação da métrica TNOAConcern (Número total de atributos do interesse). No software Ábaco TNOAConcern resulta em 27 (vinte e sete) na versão OA e 37 (trinta e sete) na versão OO, no software CDStore os resultados foram 27 (vinte e sete) na versão OA e 38 (trinta e oito) para a versão OO.

**Tabela 20 – Aplicação da métrica TNOAConcern**

	TNOAConcern
Ábaco-OA	27
Ábaco-OO	37
CDStore-OA	27
CDStore-OO	38

Na Tabela 21 são apresentados os resultados obtidos com a aplicação da métrica TLOCCConcern (Número total de linhas de código do interesse). No software Ábaco TLOCCConcern resulta em 2409 (dois mil quatrocentos e nove) na versão OA e 2330 (dois mil trezentos e trinta) na versão OO, no software CDStore os resultados foram 2409 (dois mil quatrocentos e nove) na versão OA e 2319 (dois mil trezentos e dezenove) para a versão OO.

**Tabela 21 – Aplicação da métrica TLOCCConcern**

	TLOCCConcern
Ábaco-OA	2409
Ábaco-OO	2330
CDStore-OA	2409
CDStore-OO	2319

### 3.4 – Análise dos Resultados

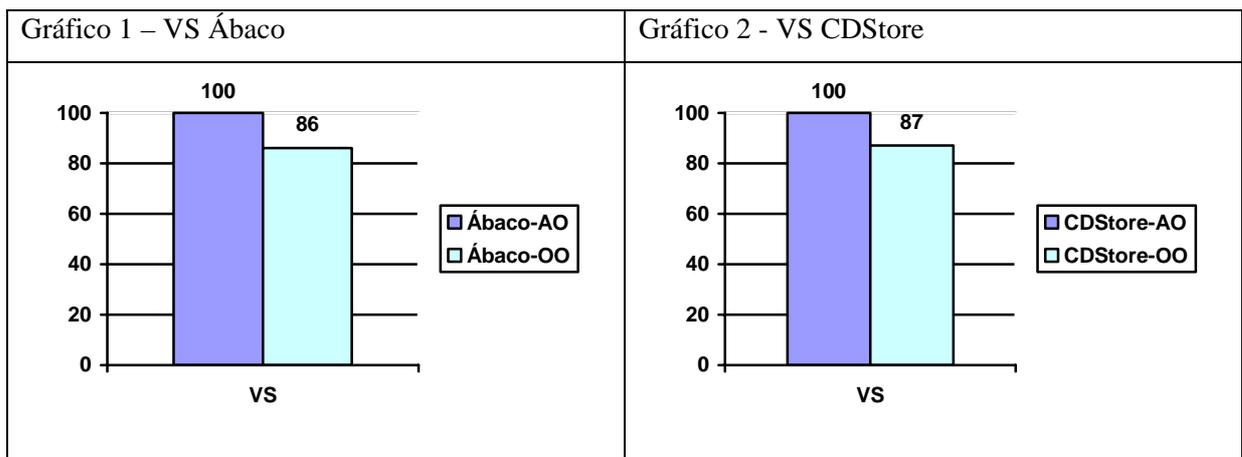
#### 3.4.1 – Métricas Tradicionais

O Gráfico 1 mostra em números percentuais o tamanho do vocabulário, do software Ábaco nas versões OA e OO, medido pela métrica VS. O software Ábaco-OA apresentou VS igual a cinquenta e sete, que no Gráfico 1 representa cem por cento, enquanto o software Ábaco-OO apresentou VS igual a quarenta e nove, que no Gráfico 1 representa em números inteiros oitenta e seis por cento, com isso é possível identificar um aumento de quatorze por cento, do tamanho do vocabulário na versão OA do software Ábaco, que pela definição da

métrica indica que o Ábaco-OO está menos espalhado, com isso mais fácil será entendê-lo para o reúso e menos componentes terão que ser alterados em um processo de manutenção.

O Gráfico 2 mostra em números percentuais o tamanho do vocabulário, do software CDStore nas versões OA e OO, medido pela métrica VS. O software CDStore-OA apresentou VS igual a quarenta e seis, que no Gráfico 2 representa cem por cento, enquanto o software CDStore-OO apresentou VS igual a quarenta, que no Gráfico 2 representa em números inteiros oitenta e sete por cento, com isso é possível identificar um aumento de treze por cento, do tamanho do vocabulário na versão OA do software CDStore, que pela definição da métrica indica que o CDStore-OO está menos espalhado, com isso mais fácil será entendê-lo para o reúso e menos componentes terão que ser alterados em um processo de manutenção.

Pela definição da métrica VS pode-se concluir que, na versão OA dos dois sistemas houve um aumento no tamanho do vocabulário aumentando a dificuldade de entender e reutilizar o código.

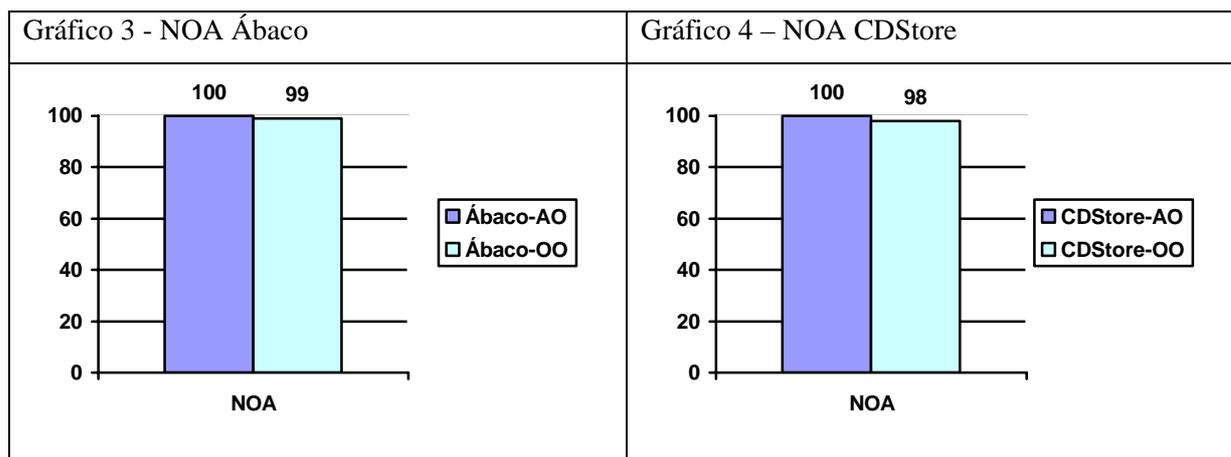


O Gráfico 3 mostra em números percentuais o número total de atributos, do software Ábaco nas versões OA e OO, medido pela métrica NOA. O software Ábaco-OA apresentou NOA igual a oitenta, que no Gráfico 3 representa cem por cento, enquanto o software Ábaco-OO apresentou NOA igual a setenta e nove, que no Gráfico 3 representa em números inteiros noventa e nove por cento, com isso é possível identificar um aumento de um por cento, no número total de atributos na versão OA do software Ábaco.

O Gráfico 4 mostra em números percentuais o número total de atributos, do software CDStore nas versões OA e OO, medido pela métrica NOA. O software CDStore-OA apresentou NOA igual a cinquenta e um, que no Gráfico 4 representa cem por cento, enquanto o software CDStore-OO apresentou NOA igual a cinquenta, que no Gráfico 4 representa em

números inteiros noventa e oito por cento, com isso é possível identificar um aumento de dois por cento, no número total de atributos na versão OA do software CDStore.

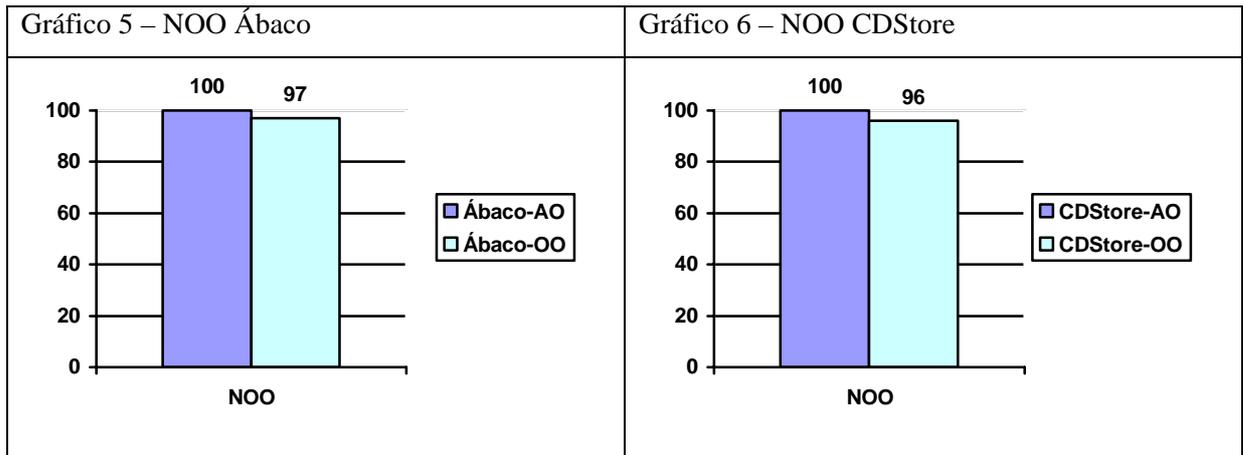
Pela definição da métrica NOA pode-se concluir que, na versão OA houve um aumento no número de atributos nos dois sistemas, tornando-os mais difícil de entender, e mais difícil de encontrar os locais que devem ser alterados.



O Gráfico 5 mostra em números percentuais o número total de operações, do software Ábaco nas versões OA e OO, medido pela métrica NOO. O software Ábaco-OA apresentou NOO igual a trezentos e vinte e nove, que no Gráfico 5 representa cem por cento, enquanto o software Ábaco-OO apresentou NOO igual a trezentos e vinte, que no Gráfico 5 representa em números inteiros noventa e sete por cento, com isso é possível identificar um aumento de três por cento, no número total de operações na versão OA do software Ábaco.

O Gráfico 6 mostra em números percentuais o número total de operações, do software CDStore nas versões OA e OO, medido pela métrica NOO. O software CDStore-OA apresentou NOO igual a duzentos e trinta e sete, que no Gráfico 6 representa cem por cento, enquanto o software CDStore-OO apresentou NOO igual a duzentos e vinte e sete, que no Gráfico 6 representa em números inteiros noventa e seis por cento, com isso é possível identificar um aumento de quatro por cento, no número total de operações na versão OA do software CDStore.

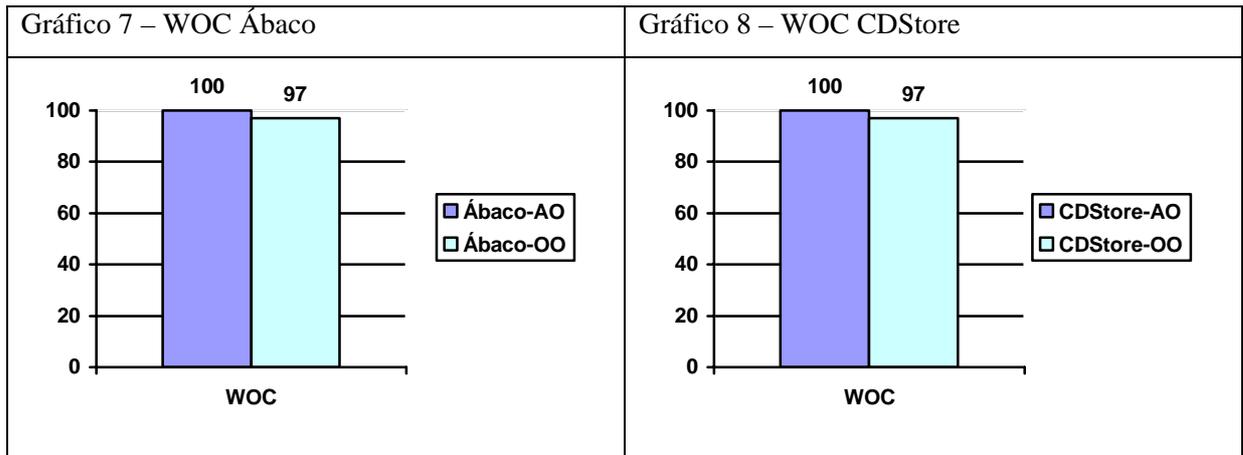
Pela definição da métrica NOO pode-se concluir que, na versão OA houve um aumento no número de operações nos dois sistemas, tornando-os mais difícil de entender, e mais difícil de encontrar os locais que devem ser alterados.



O Gráfico 7 mostra em números percentuais o peso de operação por componente, do software Ábaco nas versões OA e OO, medido pela métrica WOC. O software Ábaco-OA apresentou WOC igual a seiscentos e quatro, que no Gráfico 7 representa cem por cento, enquanto o software Ábaco-OO apresentou WOC igual a quinhentos e noventa, que no Gráfico 7 representa em números inteiros noventa e sete por cento, com isso é possível identificar um aumento de três por cento, no peso de operação por componente na versão OA do software Ábaco.

O Gráfico 8 mostra em números percentuais o peso de operação por componente, do software CDStore nas versões OA e OO, medido pela métrica WOC. O software CDStore - OA apresentou WOC igual a quatrocentos e trinta e oito, que no Gráfico 8 representa cem por cento, enquanto o software Ábaco-OO apresentou WOC igual a quatrocentos e vinte e cinco, que no Gráfico 8 representa em números inteiros noventa e sete por cento, com isso é possível identificar um aumento de três por cento, no peso de operação por componente na versão OA do software CDStore.

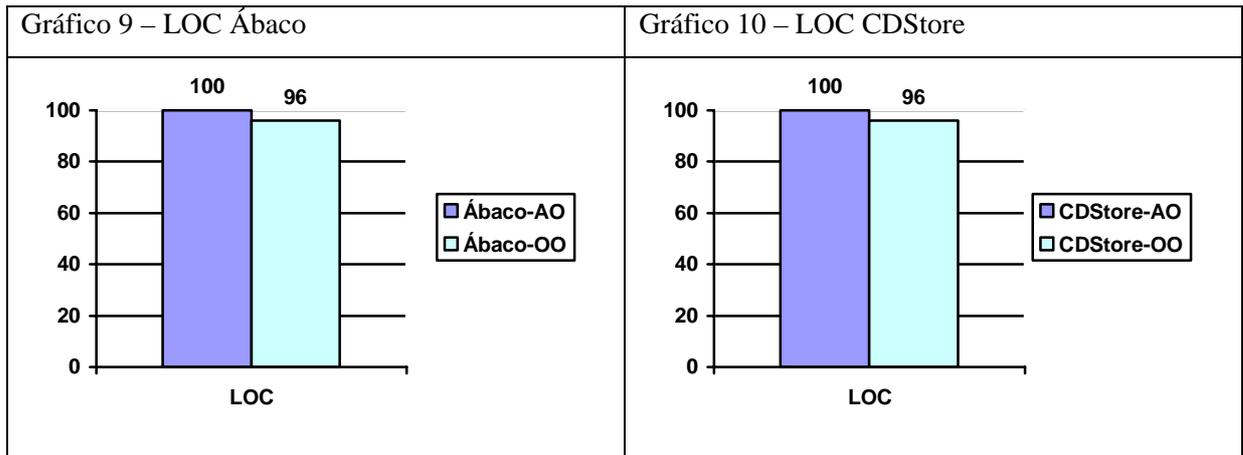
Pela definição da métrica WOC pode-se concluir que, na versão OA houve um aumento da complexidade dos dois sistemas, tornando-os mais difícil de entender, e mais difícil de realizar o processo de manutenção.



O Gráfico 9 mostra em números percentuais o número de linhas de código, do software Ábaco nas versões OA e OO, medido pela métrica LOC. O software Ábaco-OA apresentou LOC igual a quatro mil e noventa e três, que no Gráfico 9 representa cem por cento, enquanto o software Ábaco-OO apresentou LOC igual a três mil novecentos e trinta, que no Gráfico 9 representa em números inteiros noventa e seis por cento, com isso é possível identificar um aumento de quatro por cento, no número de linhas de código na versão OA do software Ábaco.

O Gráfico 10 mostra em números percentuais o número de linhas de código, do software CDStore nas versões OA e OO, medido pela métrica LOC. O software CDStore-OA apresentou LOC igual a três mil quinhentos e vinte e nove, que no Gráfico 10 representa cem por cento, enquanto o software CDStore-OO apresentou LOC igual a três mil trezentos e noventa e seis, que no Gráfico 10 representa em números inteiros noventa e seis por cento, com isso é possível identificar um aumento de quatro por cento, no número de linhas de código na versão AO do software CDStore.

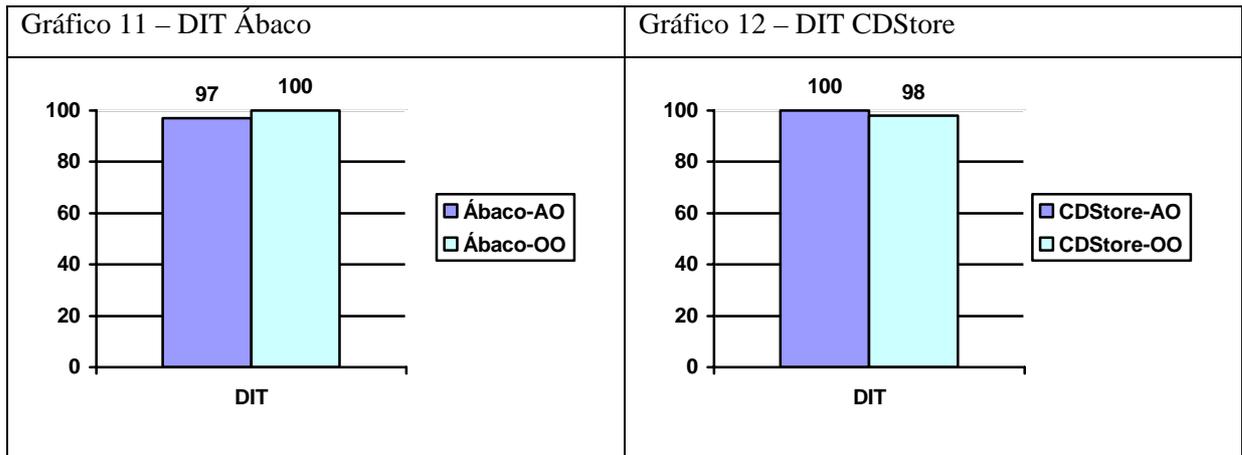
Pela definição da métrica LOC pode-se concluir que, na versão OA houve um aumento no número de linhas de código nos dois sistemas, tornando-os mais difícil de entender, e mais difícil de localizar o ponto que deve ser alterado no processo de manutenção.



O Gráfico 11 mostra em números percentuais a profundidade da árvore de herança, do software Ábaco nas versões OA e OO, medido pela métrica DIT. O software Ábaco-OA apresentou DIT igual a oitenta e três, que no Gráfico 11 representa em números inteiros noventa e sete por cento, enquanto o software Ábaco-OO apresentou DIT igual a oitenta e cinco, que no Gráfico 11 representa cem por cento. Com isso é possível identificar uma redução de três por cento, na profundidade da árvore de herança na versão OA do software Ábaco.

O Gráfico 12 mostra em números percentuais a profundidade da árvore de herança, do software CDStore nas versões OA e OO, medido pela métrica DIT. O software CDStore-OA apresentou DIT igual a sessenta e cinco, que no Gráfico 12 representa cem por cento, enquanto o software CDStore-OO apresentou DIT igual a sessenta e quatro, que no Gráfico 12 representa em números inteiros noventa e oito por cento, com isso é possível identificar um aumento de dois por cento, na profundidade da árvore de herança na versão AO do software CDStore.

Pela definição da métrica DIT pode-se concluir que, no programa Ábaco houve uma diminuição da profundidade da árvore de herança, na versão OA, no programa CDStore houve um aumento da profundidade da árvore de herança, na versão OA não sendo possível definir, qual dos paradigmas apresentam menor número da profundidade da árvore de herança.

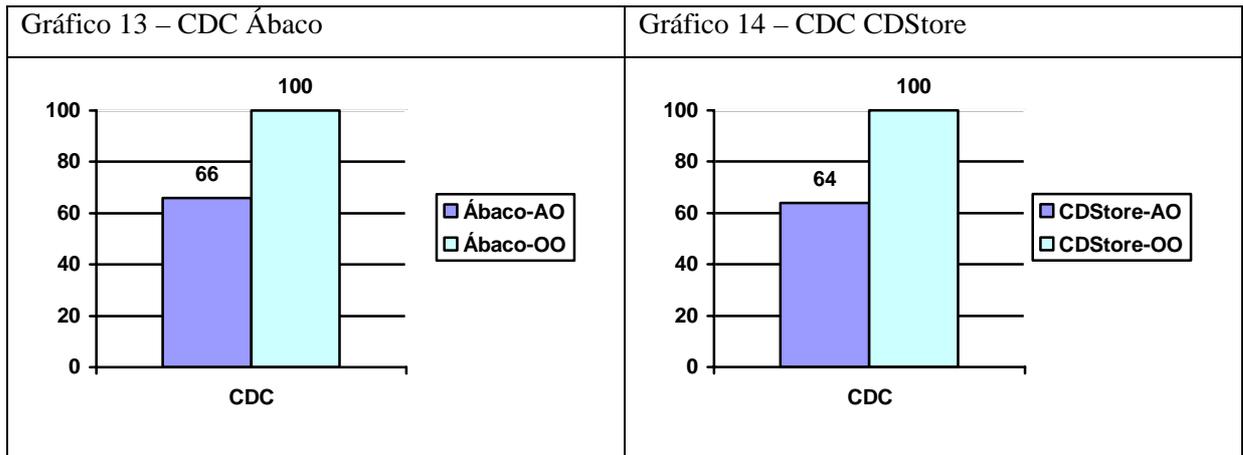


### 3.4.2 – Métricas de Separação de Interesse

O Gráfico 13 mostra em números percentuais a difusão do interesse em componentes, do software Ábaco nas versões OA e OO, medido pela métrica CDC. O software Ábaco-OA apresentou CDC igual a vinte e dois, que no Gráfico 13 representa em números inteiros sessenta e seis por cento, enquanto o software Ábaco-OO apresentou CDC igual a trinta e três, que no Gráfico 13 representa cem por cento. Com isso é possível identificar uma redução de trinta e quatro por cento, na difusão do interesse em componentes na versão OA do software Ábaco.

O Gráfico 14 mostra em números percentuais a difusão do interesse em componentes, do software CDStore nas versões OA e OO, medido pela métrica CDC. O software CDStore-OA apresentou CDC igual a vinte, que no Gráfico 14 representa em números inteiros sessenta e quatro por cento, enquanto o software CDStore-OO apresentou CDC igual a trinta e um, que no Gráfico 14 representa cem por cento. Com isso é possível identificar uma redução de trinta e seis por cento, na difusão do interesse em componentes na versão OA do software CDStore.

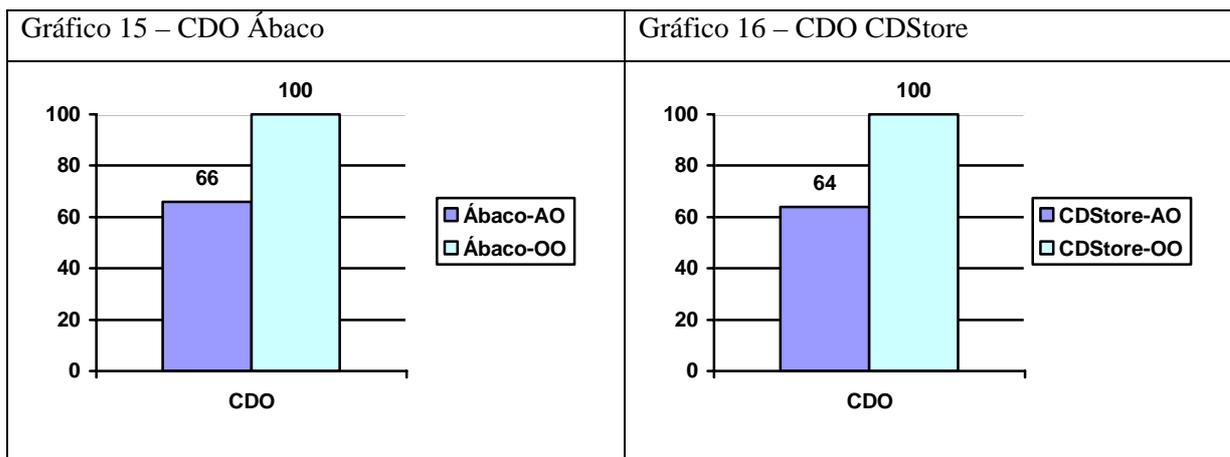
Pela definição da métrica CDC pode-se concluir que, na versão OA houve uma melhora significativa na difusão do interesse em componentes, esses encontram-se menos espalhados que na versão OO, os tornando assim mais fácil de entender.



O Gráfico 15 mostra em números percentuais a difusão do interesse em operações, do software Ábaco nas versões OA e OO, medido pela métrica CDO. O software Ábaco-OA apresentou CDO igual a cento e quarenta e sete, que no Gráfico 15 representa em números inteiros noventa e seis por cento, enquanto o software Ábaco-OO apresentou CDO igual a cento e cinquenta e dois, que no Gráfico 15 representa cem por cento. Com isso é possível identificar uma redução de quatro por cento, na difusão do interesse em operações na versão OA do software Ábaco.

O Gráfico 16 mostra em números percentuais a difusão do interesse em operações, do software CDStore nas versões OA e OO, medido pela métrica CDO. O software CDStore-OA apresentou CDO igual a cento e quarenta e sete, que no Gráfico 16 representa em números inteiros noventa e quatro por cento, enquanto o software CDStore-OO apresentou CDO igual a cento e cinquenta e seis, que no Gráfico 16 representa cem por cento. Com isso é possível identificar uma redução de seis por cento, na difusão do interesse em operações na versão AO do software CDStore.

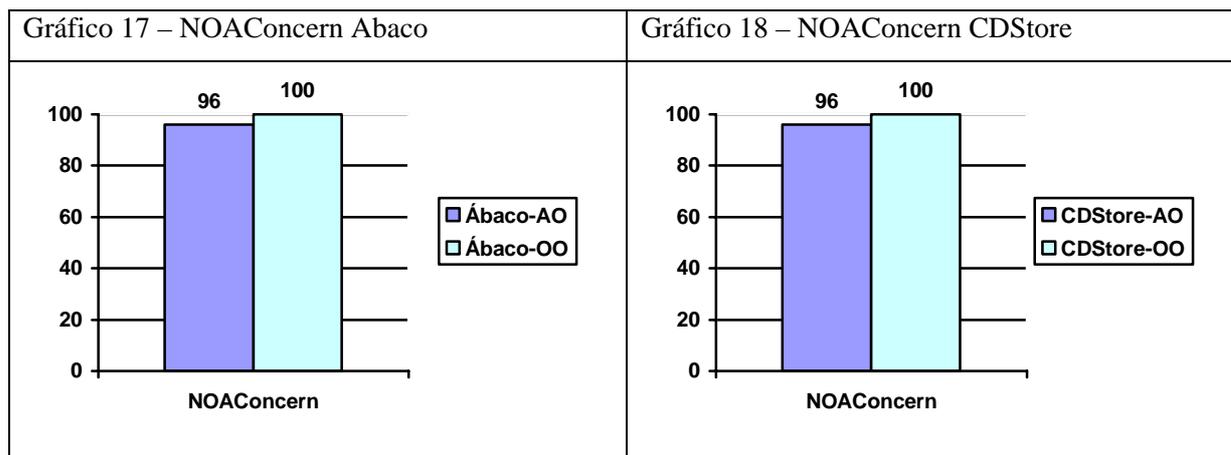
Pela definição da métrica CDO pode-se concluir que, na versão OA houve uma melhora significativa na difusão do interesse em operações, tornando assim o interesse mais fácil de ser entendido e ser reutilizado em um processo de manutenção.



O Gráfico 17 mostra em números percentuais o número de atributos de interesse, do software Ábaco nas versões OA e OO, medido pela métrica NOAConcern. O software Ábaco-OA apresentou NOAConcern igual a vinte e sete, que no Gráfico 17 representa cem por cento, enquanto o software Ábaco-OO apresentou NOAConcern igual a vinte e seis, que no Gráfico 17 representa em números inteiros noventa e seis por cento. Com isso é possível identificar um aumento de quatro por cento, no número de atributos de interesse na versão OA do software Ábaco.

O Gráfico 18 mostra em números percentuais o número de atributos de interesse, do software CDStore nas versões OA e OO, medido pela métrica NOAConcern. O software CDStore-OA apresentou NOAConcern igual a vinte e sete, que no Gráfico 18 representa cem por cento, enquanto o software Ábaco-OO apresentou NOAConcern igual a vinte e seis, que no Gráfico 18 representa em números inteiros noventa e seis por cento. Com isso é possível identificar um aumento de quatro por cento, no número de atributos de interesse na versão OA do software CDStore.

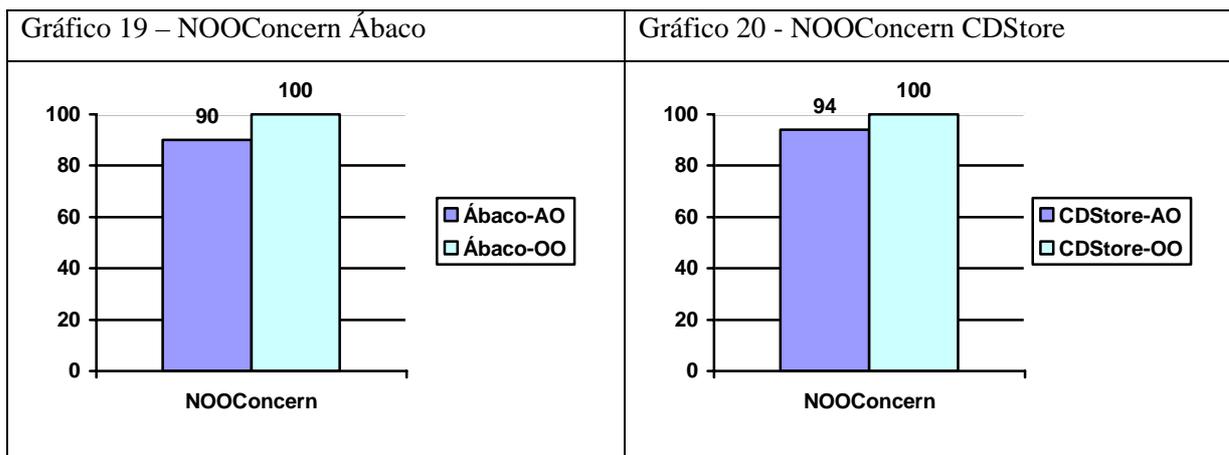
Pela definição da métrica NOAConcern pode-se concluir que, na versão OA houve uma melhora na difusão do interesse em atributos, esses encontram-se menos espalhados que na versão OO.



O Gráfico 19 mostra em números percentuais uma comparação entre o número de operações de interesse, do software Ábaco nas versões OA e OO, medido pela métrica NOOConcern. O software Ábaco-OA apresentou NOOConcern igual a cento e quarenta e sete, que no Gráfico 19 representa cem por cento, enquanto o software Ábaco-OO apresentou NOOConcern igual a cento e trinta e três, que no Gráfico 19 representa em números inteiros noventa por cento. Com isso é possível identificar um aumento de dez por cento, no número de operações de interesse na versão OA do software Ábaco.

O Gráfico 20 mostra em números percentuais uma comparação entre o número de operações de interesse, do software CDStore nas versões OA e OO, medido pela métrica NOOConcern. O software CDStore-OA apresentou NOOConcern igual a cento e quarenta e sete, que no Gráfico 20 representa cem por cento, enquanto o software CDStore-OO apresentou NOOConcern igual a cento e trinta e nove, que no Gráfico 20 representa em números inteiros noventa e quatro por cento. Com isso é possível identificar um aumento de seis por cento, no número de operações de interesse na versão OA do software CDStore.

Pela definição da métrica NOOConcern pode-se concluir que, na versão OA houve uma melhora na difusão do interesse em operações, esses encontram-se menos espalhados que na versão OO.

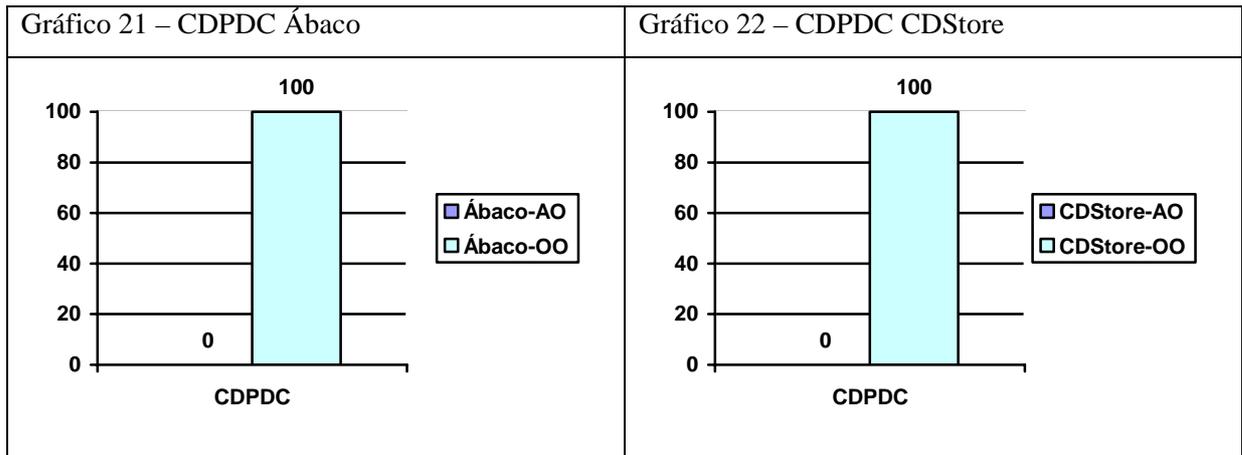


### 3.4.3 – Métricas Propostas

O Gráfico 21 mostra em números percentuais uma comparação entre a difusão do interesse em componentes parcialmente dedicados, do software Ábaco nas versões OA e OO, medido pela métrica CDPDC. O software Ábaco-OA apresentou CDPDC igual a 0 (zero), enquanto o software Ábaco-OO apresentou CDPDC igual a 20 (vinte), que no Gráfico 21 representa 100 % (cem por cento). Com isso é possível identificar que não há difusão do interesse em componentes parcialmente dedicados na versão OA do software Ábaco.

O Gráfico 22 mostra em números percentuais uma comparação entre a difusão do interesse em componentes parcialmente dedicados, do software CDStore nas versões OA e OO, medido pela métrica CDPDC. O software CDStore-OA apresentou CDPDC igual a zero, enquanto o software CDStore-OO apresentou CDPDC igual a 20 (vinte), que no Gráfico 22 representa 100% (cem por cento). Com isso é possível identificar que não há difusão do interesse em componentes parcialmente dedicados na versão OA do software CDStore.

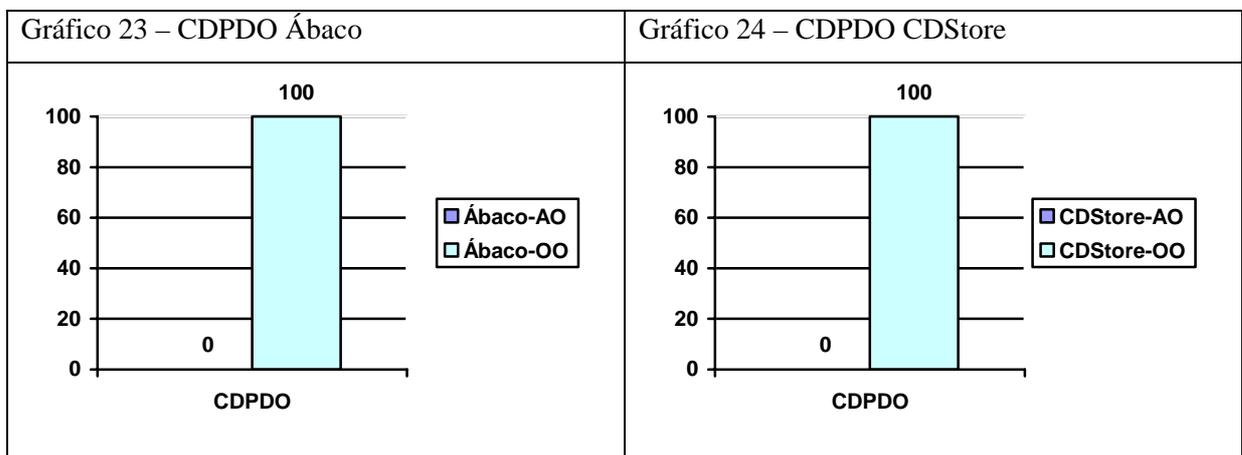
Pela definição da métrica CDPDC pode-se concluir que, na versão OA não há componentes parcialmente dedicados a implementação de interesses, já na versão OO há um espalhamento do interesse.



O Gráfico 23 demonstra em números percentuais uma comparação entre a difusão do interesse em operações parcialmente dedicadas, do software Ábaco nas versões OA e OO, medido pela métrica CDPDO. O software Ábaco-OA apresentou CDPDO igual a 0 (zero), enquanto o software Ábaco-OO apresentou CDPDO igual a 19 (dezenove), que no Gráfico 23 representa 100% (cem por cento). Com isso é possível identificar que não há difusão do interesse em operações parcialmente dedicadas na versão OA do software Ábaco.

O Gráfico 24 demonstra em números percentuais uma comparação entre a difusão do interesse em operações parcialmente dedicadas, do software CDStore nas versões OA e OO, medido pela métrica CDPDO. O software CDStore-OA apresentou CDPDO igual a 0 (zero), enquanto o software CDStore-OO apresentou CDPDO igual a 17 (dezesete), que no Gráfico 24 representa 100% (cem por cento). Com isso é possível identificar que não há difusão do interesse em operações parcialmente dedicadas na versão OA do software CDStore.

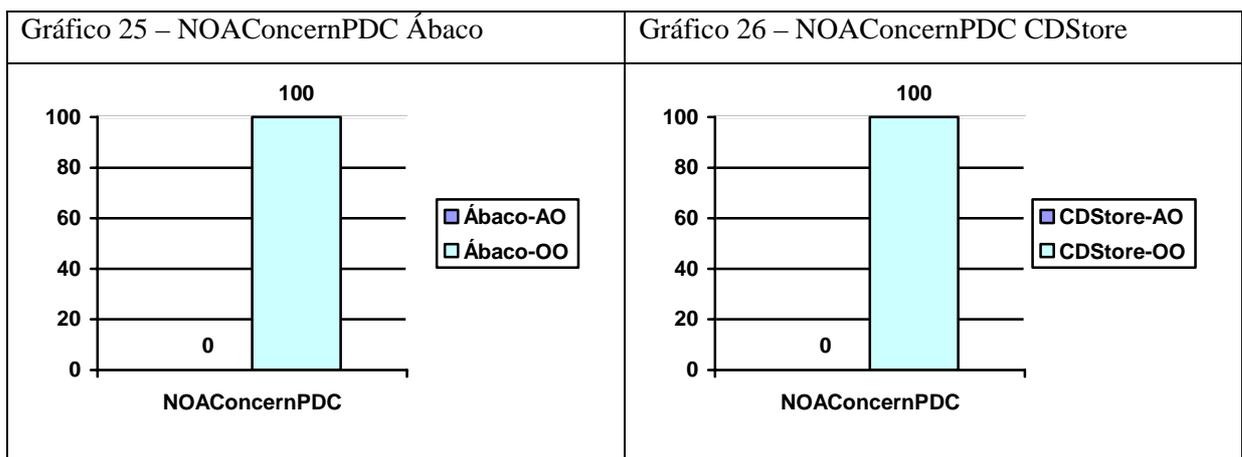
Pela definição da métrica CDPDO pode-se concluir que, na versão OA não há operações parcialmente dedicadas à implementação de interesses, já na versão OO há um espalhamento das operações do interesse.



O Gráfico 25 demonstra em números percentuais uma comparação entre o número de atributos do interesse em componentes parcialmente dedicados, do software Ábaco nas versões OA e OO, medido pela métrica NOAConcernPDC. O software Ábaco-OA apresentou NOAConcernPDC igual a 0 (zero), enquanto o software Ábaco-OO apresentou NOAConcernPDC igual a 11 (onze), que no Gráfico 25 representa 100% (cem por cento). Com isso é possível identificar que não há atributos do interesse em componentes parcialmente dedicados na versão OA do software Ábaco.

O Gráfico 26 demonstra em números percentuais uma comparação entre o número de atributos do interesse em componentes parcialmente dedicados, do software CDStore nas versões OA e OO, medido pela métrica NOAConcernPDC. O software CDStore-OA apresentou NOAConcernPDC igual a 0 (zero), enquanto o software CDStore-OO apresentou CDPDO igual a 12 (doze), que no Gráfico 26 representa 100% (cem por cento). Com isso é possível identificar que não há atributos do interesse em componentes parcialmente dedicados na versão OA do software CDStore.

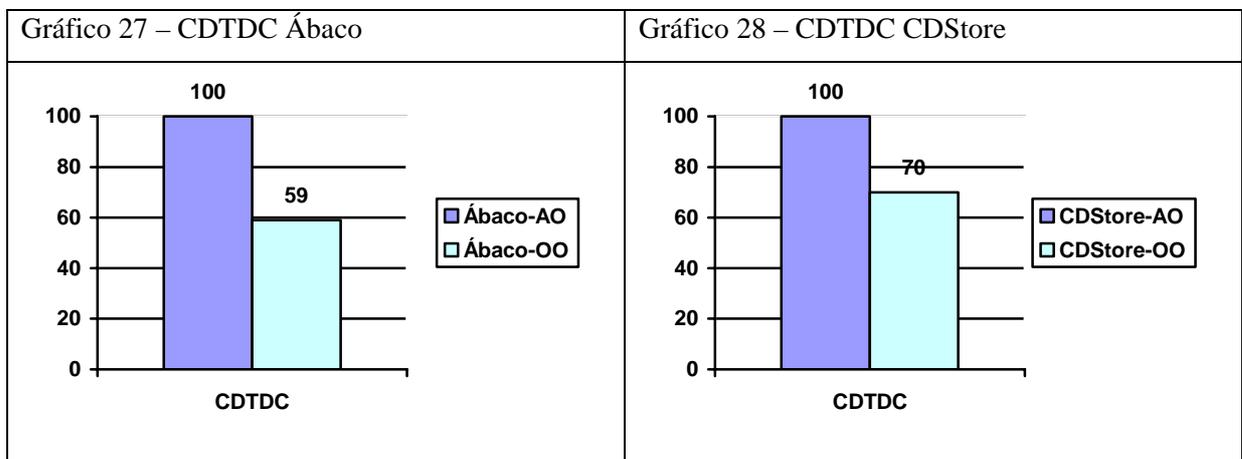
Pela definição da métrica NOAConcernPDC pode-se concluir que, na versão OA não há atributos em componentes parcialmente dedicados a implementação de interesses, já na versão OO há um espalhamento dos atributos do interesse em componentes parcialmente dedicados.



O Gráfico 27 demonstra em números percentuais uma comparação entre a difusão do interesse em componentes totalmente dedicados, do software Ábaco nas versões OA e OO, medido pela métrica CDTDC. O software Ábaco-OA apresentou CDTDC igual a 22 (vinte e dois), que no Gráfico 27 representa 100% (cem por cento), enquanto o software Ábaco-OO apresentou CDTDC igual a 13 (treze), que no Gráfico 27 representa 70% (setenta por cento).

O Gráfico 28 demonstra em números percentuais uma comparação entre a difusão do interesse em componentes totalmente dedicados, do software CDStore nas versões OA e OO, medido pela métrica CDTDC. O software CDStore-OA apresentou CDTDC igual a 20 (vinte), que no Gráfico 28 representa 100% (cem por cento), enquanto o software CDStore-OO apresentou CDPDO igual a 14 (quatorze), que no Gráfico 28 representa 70% (setenta por cento).

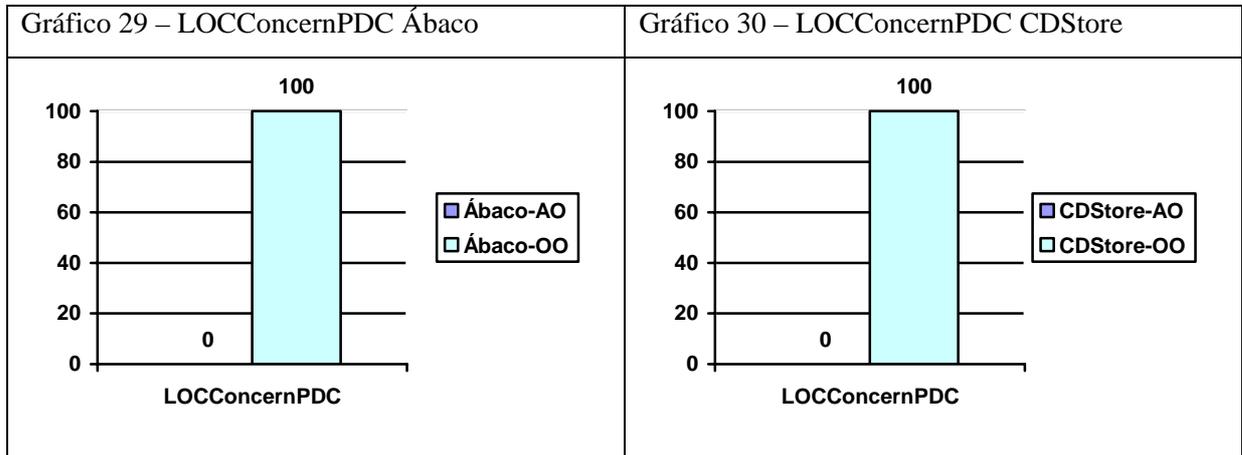
A métrica CDTDC será utilizada, como apoio para a interpretação da regra heurística R04.



O Gráfico 29 demonstra em números percentuais uma comparação entre o número de linhas de código do interesse em componentes parcialmente dedicados, do software Ábaco nas versões OA e OO, medido pela métrica LOCCConcernPDC. O software Ábaco-OA apresentou LOCCConcernPDC igual a 0 (zero), enquanto o software Ábaco-OO apresentou LOCCConcernPDC igual a 57 (cinquenta e sete), que no Gráfico 29 representa 100% (cem por cento). Com isso é possível identificar que não há linhas de código do interesse em componentes parcialmente dedicados na versão OA do software Ábaco.

O Gráfico 30 demonstra em números percentuais uma comparação entre o número de linhas de código do interesse em componentes parcialmente dedicados, do software CDStore nas versões OA e OO, medido pela métrica LOCCConcernPDC. O software CDStore-OA apresentou LOCCConcernPDC igual a 0 (zero), enquanto o software CDStore-OO apresentou LOCCConcernPDC igual a 51 (cinquenta e um), que no Gráfico 30 representa 100% (cem por cento). Com isso é possível identificar que não há linhas de código do interesse em componentes parcialmente dedicados na versão OA do software CDStore.

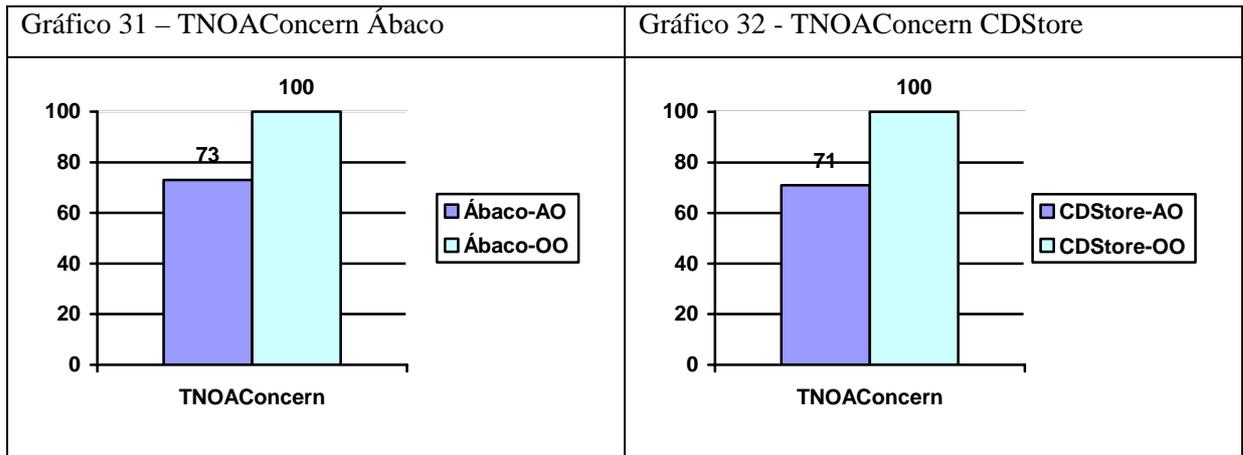
Pela definição da métrica LOCConcernPDC pode-se concluir que, na versão OA não há linhas de código do interesse em componentes parcialmente dedicados a implementação de interesses, já na versão OO há um entrelaçamento das linhas de código do interesse em componentes parcialmente dedicados.



O Gráfico 31 demonstra em números percentuais uma comparação entre o total de atributos do interesse, do software Ábaco nas versões OA e OO, medido pela métrica TNOAConcern. O software Ábaco-OA apresentou TNOAConcern igual a 27 (vinte e sete), enquanto o software Ábaco-OO apresentou TNOAConcern igual a 37 (trinta e sete), que no Gráfico 31 representa 100% (cem por cento).

O Gráfico 32 demonstra em números percentuais uma comparação entre o total de atributos do interesse, do software CDStore nas versões OA e OO, medido pela métrica TNOAConcern. O software CDStore-OA apresentou TNOAConcern igual a 27 (vinte e sete), enquanto o software CDStore-OO apresentou TNOAConcern igual a 38 (trinta e oito), que no Gráfico 30 representa 100% (cem por cento).

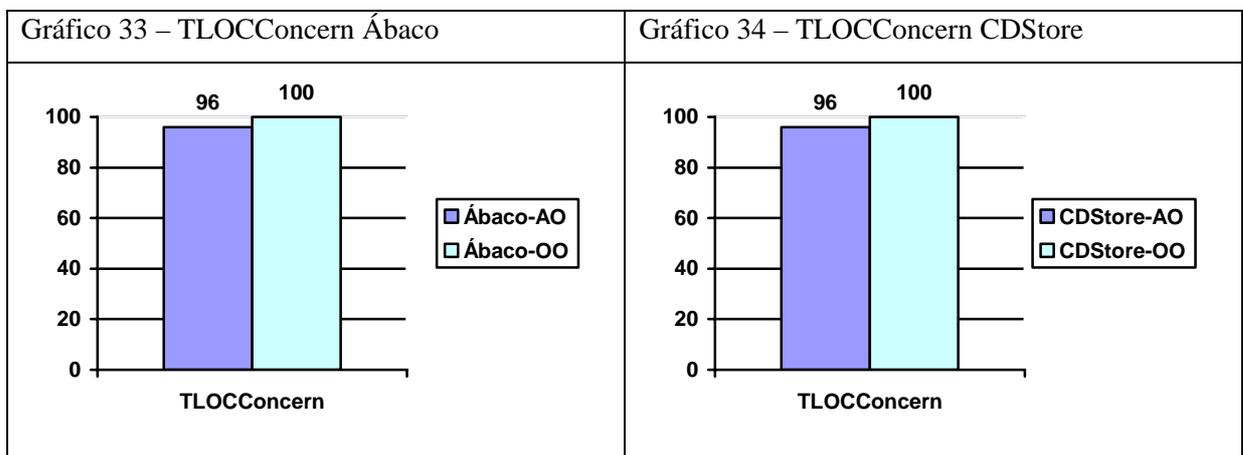
A métrica TNOAConcern será utilizada, como apoio para a interpretação da regra heurística R05.



O Gráfico 33 demonstra em números percentuais uma comparação entre o total de linhas de código do interesse, do software Ábaco nas versões OA e OO, medido pela métrica TLOCCConcern. O software Ábaco-OA apresentou TLOCCConcern igual a 2409 (dois mil quatrocentos e nove), que no Gráfico 33 representa 100% (cem por cento), enquanto o software Ábaco-OO apresentou TLOCCConcern igual a 2330 (dois mil trezentos e trinta), que no Gráfico 33 representa 96% (cem por cento).

O Gráfico 34 demonstra em números percentuais uma comparação entre o total de linhas de código do interesse, do software CDStore nas versões OA e OO, medido pela métrica TLOCCConcern. O software CDStore-OA apresentou TLOCCConcern igual a 2409 (dois mil quatrocentos e nove), que no Gráfico 34 representa 100% (cem por cento), enquanto o software Ábaco-OO apresentou TLOCCConcern igual a 2319 (dois mil trezentos e dezenove), que no Gráfico 34 representa 96% (cem por cento).

A métrica TLOCCConcern será utilizada, como apoio para a interpretação da regra heurística R07.



### 3.5 – Aplicação das Regras Heurísticas Propostas

Nesta Seção serão interpretados os dados obtidos com a aplicação das métricas propostas na Seção 2.3, com o apoio das regras heurísticas propostas na mesma Seção. Os resultados das métricas sem o apoio dessas regras é pouco dedutivo e de difícil interpretação.

Na Tabela 22 são apresentados os resultados obtidos com a aplicação das regras heurísticas comparativas R01 e R02, para isso serão utilizadas as métricas CDO (difusão do interesse em operações), NOOConcern (número de operações de interesse) e CDPDO (Difusão do interesse em operações parcialmente dedicadas). Difusão do interesse em operações parcialmente dedicadas.

No software Ábaco-OA o valor da métrica CDO é igual a 147 (cento e quarenta e sete) e o valor da métrica NOOConcern também representa 147 (cento e quarenta e sete). O valor obtido com a aplicação da métrica NOOConcern dividido pelo valor obtido com a aplicação da métrica CDC é igual a 1 (um), o que indica que 100 % (cem por cento) das operações dedicadas à do interesse avaliado está em componentes totalmente dedicados. O valor obtido com a aplicação da métrica CDPDO é igual a 0 (zero), conseqüentemente a divisão do valor da métrica CDPDO pelo valor da métrica CDO e a divisão do valor da métrica CDPDO pelo valor da métrica NOOConcern, também são iguais a 0 (zero), o que indica que não há operações em componentes parcialmente dedicados, isso leva a conclusão que referente a operações relacionadas ao interesse de persistência no software Ábaco-OA a separação de interesse é total.

No software Ábaco-OO o valor obtido com a aplicação da métrica CDO é igual a 152 (cento e cinquenta e dois) e o valor obtido com a aplicação da métrica NOOConcern é 133 (cento e trinta e três). O valor da métrica NOOConcern dividido pelo valor da métrica CDC é igual a 0.87 (zero ponto oitenta e sete), o que indica que 87% (oitenta e sete por cento) dos componentes relacionados ao interesse avaliado estão em componentes totalmente dedicados. O valor obtido com a aplicação da métrica CDPDO é igual a 19 (dezenove), o valor da métrica CDPDO dividido pelo valor da métrica CDO é igual a 0.13 (zero ponto treze), o que indica que 13% (treze por cento) do total de operações do interesse avaliado está “entrelaçado” no código base. Para essa aplicação o valor da métrica CDPDO dividido pelo valor da métrica NOOConcern é igual 0.14 (zero ponto quatorze), o que indica que as operações parcialmente dedicadas representam 14% (quatorze por cento) das operações totalmente dedicadas, isso leva a conclusão que referente a operações relacionadas ao

interesse de persistência no software Ábaco-OO as operações do interesse estão 13% (treze por cento) entrelaçadas.

No software CDStore-OA o valor obtido com a aplicação da métrica CDO é igual a 147 (cento e quarenta e sete) e o valor obtido com aplicação da métrica NOOConcern também representa 147 (cento e quarenta e sete). O valor da métrica NOOConcern dividido por CDC é igual a 1 (um), o que indica que 100% (cem por cento) das operações dedicadas à implementação do interesse avaliado, estão em componentes totalmente dedicados. O valor da métrica CDPDO é igual a 0 (zero) conseqüentemente a divisão do valor da métrica CDPDO dividido pelo valor da métrica CDO e o valor da métrica CDPDO dividido pelo valor da métrica NOOConcern também são iguais a 0 (zero), o que indica que não a operações em componentes parcialmente dedicados, isso leva a conclusão que, referente a operações relacionadas ao interesse de persistência no software CDStore-OA a separação de interesse é total.

No software CDStore-OO, o valor obtido com a aplicação da métrica CDO é igual a 156 (cento e cinquenta e seis) e o valor obtido com a aplicação da métrica NOOConcern é 133 (cento e trinta e três). O valor da métrica NOOConcern dividido pelo valor da métrica CDC é igual a 0.89 (zero ponto oitenta e nove), o que indica que 89% (oitenta e nove por cento) dos componentes relacionados ao interesse avaliado, estão em componentes totalmente dedicados. O valor obtido com a aplicação da métrica CDPDO é igual a 17 (dezessete), o valor da métrica CDPDO dividido pelo valor da métrica CDO é igual a 0.1 (zero ponto um), o que indica que 10% (dez por cento) do total de operações do interesse avaliado estão “entrelaçadas” no código base. Para esta aplicação a divisão do valor da métrica CDPDO pelo valor da métrica NOOConcern é igual 0.12 (zero ponto doze), o que indica que as operações parcialmente dedicadas representam 12% (doze por cento) das operações totalmente dedicadas, isso leva a conclusão que referente as operações relacionadas ao interesse de persistência no software CDStore-OO as operações do interesse estão 12% (doze por cento) entrelaçadas.

Pelas definições das regras heurísticas R01 e R02, é possível concluir que nas duas aplicações OO, os resultados obtidos demonstram um espalhamento das operações do interesse, o que não ocorre na versão OA.

**Tabela 22 - Aplicação das regras Heurísticas Comparativas R01 e R02**

Regras Heurísticas Comparativas R01 e R02						
Software	CDO	NOOConcern	NOOConcern/ CDO	CDPDO	CDPDO/ CDO	CDPDO/ NOOConcern
Ábaco_OA	147	147	1	0	0	0
Ábaco_OO	152	133	0,87	19	0,13	0,14
CDStore_OA	147	147	1	0	0	0
CDStore_OO	156	139	0,89	17	0,10	0,12

Na Tabela 23 são apresentados os resultados obtidos com a aplicação das regras heurísticas comparativas R03 e R04, para isso serão utilizadas as métricas CDC (Difusão do Interesse em Componentes), CDTDC (Difusão do Interesse em Componentes Totalmente Dedicados) e CDPDC (Difusão do Interesse em Componentes Parcialmente Dedicados).

No software Ábaco-OA o valor obtido com a aplicação da métrica CDC é igual a 22 (vinte e dois) e o valor da métrica CDTDC também representa 22 (vinte e dois). A divisão do valor da métrica CDTDC pelo valor da métrica CDC é igual a 1 (um), o que indica que 100% (cem por cento) dos componentes dedicados à implementação do interesse são totalmente dedicados. O valor obtido com a aplicação da métrica CDPDC é igual a 0 (zero) conseqüentemente o valor da métrica CDPDC dividido pelo valor da métrica CDC e o valor da métrica CDPDC dividido pelo valor da métrica CDTDC também são iguais a 0 (zero), o que indica que não há componentes parcialmente dedicados, isso leva a conclusão que referente à componentes do interesse de persistência no software Ábaco-OA a separação de interesse é total.

No software Ábaco-OO o valor obtido com a aplicação da métrica CDC é igual a 33 (trinta e três) e o valor da métrica CDTDC é 13 (treze). O valor da métrica CDTDC dividido pelo valor da métrica CDC é igual a 0.39 (zero ponto trinta em nove), o que indica que 39% (trinta e nove por cento) dos componentes são totalmente dedicados. O valor da métrica CDPDO é igual a 20 (vinte). O valor da métrica CDPDC dividido pelo valor da métrica CDC é igual a 0.60 (zero ponto sessenta), o que indica que 60% (sessenta por cento) do total de componentes do interesse avaliado está “espalhado” no código base. Para esta aplicação o valor da métrica CDPDC dividido pelo valor da métrica CDTDC é igual a 1.53 (um ponto cinquenta e três), o que indica que os componente parcialmente dedicadas representam 153% (cento e cinquenta e três por cento) dos componentes totalmente dedicados. Isso leva a

conclusão que referente à persistência no software Ábaco-OO os componentes do interesse estão 39% (trinta e nove por cento) “espalhados”.

No software CDStore-AO o valor obtido com a aplicação da métrica CDC é igual a 20 (vinte) e o valor da métrica CDTDC também é 20 (vinte). O valor da métrica CDTDC dividido pelo valor da métrica CDC é igual a 1 (um), o que indica que 100% (cem por cento) dos componentes dedicados à implementação do interesse são totalmente dedicados. O valor da métrica CDPDC é igual a 0 (zero) conseqüentemente a divisão do valor da métrica CDPDC pelo valor da métrica CDC e o valor da métrica CDPDC dividido pelo valor da métrica CDTDC também são iguais a 0 (zero), o que indica que não há componentes parcialmente dedicados. Isso leva a conclusão que referente à componentes do interesse de persistência no software CDStore-OA a separação de interesse é total.

No software CDStore-OO o valor obtido com a aplicação da métrica CDC é igual a 31 (trinta e um) e o valor obtido com a aplicação da métrica CDTDC é 14 (quatorze), o valor da métrica CDTDC dividido pelo valor da métrica CDC é igual a 0.45 (zero ponto quarenta e cinco), o que indica que 45% (quarenta e cinco por cento) dos componentes são totalmente dedicados. O valor da métrica CDPDO é igual a 20 (vinte), e o valor da métrica CDPDC dividido pelo valor da métrica CDC é igual a 0.55 (zero ponto cinquenta e cinco), o que indica que 55% (cinquenta e cinco por cento) do total de componentes do interesse avaliado está “espalhado” no código base. Para esta aplicação o valor da métrica CDPDC dividido pelo valor da métrica CDTDC é igual 1.21 (um ponto vinte um), o que indica que os componentes parcialmente dedicados representam 121% (cento e vinte e um por cento) dos componentes totalmente dedicados. Isso leva a conclusão que referente a persistência no software CDStore-OO os componentes do interesse estão 45% (quarenta e cinco por cento) “espalhados”.

Pelas definições das regras heurísticas R03 e R04, é possível concluir que nas duas aplicações OO, os resultados obtidos demonstram um espalhamento dos componentes do interesse, o que não ocorre na versão OA.

**Tabela 23 - Aplicação das regras Heurísticas Comparativas R03 e R04**

Regras Heurísticas Comparativas R03 e R04						
Software	CDC	CDTDC	CDTDC/ CDC	CDPDC	CDPDC/ CDC	CDPDC/ CDTDC
Ábaco_OA	22	22	1	0	0	0
Ábaco_OO	33	13	0,39	20	0,60	1,53
CDStore_OA	20	20	1	0	0	0
CDStore_OO	31	14	0,45	17	0,55	1,21

Na Tabela 24 são apresentados os resultados obtidos com a aplicação das regras heurísticas comparativas R05 e R06, para isso serão utilizadas as métricas TNOAConcern (Número total de atributos do interesse), NOAConcern (número de atributos do interesse) e NOAConcernPDC (Número de atributos do interesse em componentes parcialmente dedicados).

No software Ábaco-OA o valor obtido com a aplicação da métrica TNOAConcern é igual a 27 (vinte e sete) e o valor obtido com a aplicação da métrica NOAConcern também representa 27 (vinte e sete). O valor da métrica NOAConcern dividido pelo valor da métrica TNOAConcern é igual a (1) um, o que indica que 100% (cem por cento) dos atributos estão em componentes totalmente dedicados à implementação do interesse. O valor da métrica NOAConcernPDC é igual a 0 (zero) conseqüentemente o valor da métrica NOAConcernPDC dividido pelo valor da métrica TNOAConcern e o valor da métrica NOAConcernPDC dividido pelo valor da métrica NOAConcern também são iguais a 0 (zero), o que indica que não há atributos do interesse em componentes parcialmente dedicados. Isso leva a conclusão que no referente à atributos do interesse de persistência. No software Ábaco-OA a separação de interesse é total.

No software Ábaco-OO o valor obtido com a aplicação da métrica TNOAConcern é igual a 37 (trinta e sete) e o valor obtido com a aplicação da métrica NOAConcern é 26 (vinte e seis), a divisão do valor da métrica NOAConcern pelo valor da métrica TNOAConcern é igual a 0.7 (zero ponto sete), o que indica que 70% (setenta por cento) dos atributos estão em componentes totalmente dedicados. O valor obtido com a aplicação da métrica NOAConcernPDC é igual a 11 (onze), e a divisão do valor obtido com a aplicação da métrica NOAConcernPDC pelo valor da métrica TNOAConcern é igual a 0.30 (zero ponto trinta), o que indica que 30% (trinta por cento) do total de atributos do interesse avaliado estão “espalhados” no código base. Para esta aplicação o valor obtido com a aplicação da métrica NOAConcernPDC dividido pelo valor da métrica NOAConcern é igual 0.42 (zero ponto quarenta e dois), o que indica que os atributos dedicados ao interesse em de componentes parcialmente dedicados representam 42% (quarenta e dois por cento) dos atributos dos componentes totalmente dedicados. Isso leva a conclusão que referente à persistência no software Ábaco-OO os atributos do interesse estão 30% (trinta por cento) “espalhados”. No software CDStore-AO o valor obtido com a aplicação da métrica TNOAConcern é igual a 27 (vinte e sete) e o valor da métrica NOAConcern também representa 27 (vinte e sete), o valor da métrica NOAConcern dividido pelo valor da métrica TNOAConcern é igual a 1 (um), o que indica que 100% (cem por cento) dos atributos estão em componentes totalmente

dedicados à implementação do interesse. O valor obtido com a aplicação da métrica NOAConcernPDC é igual a 0 (zero), conseqüentemente o valor da métrica NOAConcernPDC dividido pelo valor da métrica TNOAConcernPDC, e pelo valor da métrica NOAConcernPDC dividido pelo valor da métrica NOAConcern também são iguais a 0 (zero), o que indica que não atributos do interesse em componentes parcialmente dedicados. Isso leva a conclusão que referente à atributos do interesse de persistência no software CDStore-OA a separação de interesse é total.

No software CDStore-OO o valor obtido com a aplicação da métrica TNOAConcern é igual a 38 (trinta e oito) e o valor obtido com a aplicação da métrica NOAConcern é 26 (vinte e seis), a divisão do valor da métrica NOAConcern pelo valor da métrica TNOAConcern é igual a 0.68 (zero ponto sessenta e oito), o que indica que 68% (sessenta e oito por cento) dos atributos estão em componentes totalmente dedicados. O valor obtido com a aplicação da métrica NOAConcernPDC é igual a 12 (doze), e a divisão do valor da métrica NOAConcernPDC pelo valor da métrica TNOAConcern é igual a 0.32 (zero ponto trinta e dois), o que indica que 32% (trinta e dois por cento) do total de atributos do interesse avaliado estão “espalhados” no código base. Para esta aplicação o valor da métrica NOAConcernPDC dividido pelo valor da métrica NOAConcern é igual a 0.46 (zero ponto quarenta e seis), o que indica que os atributos dedicados ao interesse em de componentes parcialmente dedicados representam 46% (quarenta e seis por cento) dos atributos dos componentes totalmente dedicados. Isso leva a conclusão que referente à persistência no software CDStore-OO os atributos do interesse estão 32% (Trinta e dois por cento) “espalhados”.

Pelas definições das regras heurísticas R05 e R06, é possível concluir que nas duas aplicações OO, os resultados obtidos demonstram um espalhamento dos atributos de interesse, que na versão OA não ocorre.

**Tabela 24 - Aplicação das regras Heurísticas Comparativas R05 e R06**

Regras Heurísticas Comparativas R05 e R06						
Software	TNOA Concern	NOA Concern	NOAConcern/ TNOAConcern	NOAConcern PDC	NOAConcern PDC/ TNOAConcern	NOAConcern PDC/ NOAConcern
Ábaco_OA	27	27	1	0	0	0
Ábaco_OO	37	26	0,7	11	0,3	0,42
CDStore_OA	27	27	1	0	0	0
CDStore_OO	38	26	0,68	12	0,32	0,46

Na Tabela 25 são apresentados os resultados obtidos com a aplicação das regras heurísticas comparativas R07 e R08, para isso serão usadas as métricas LOCCConcern (número de linhas de código do interesse), TLOCCConcern (número total de linhas de código do interesse) e LOCCConcernPDC (Número de linhas de código do interesse em componentes parcialmente dedicados).

No software Ábaco-OA, o valor obtido com a aplicação da métrica TLOCCConcern é igual a 2409 (dois mil quatro centos e nove) e o valor obtido com a aplicação da métrica LOCCConcern também é igual a 2409 (dois mil quatro centos e nove), o valor da métrica LOCCConcern dividido pelo valor da métrica TLOCCConcern é igual a 1 (um), o que indica que 100% (cem por cento) das linhas do interesse estão em componentes totalmente dedicados à implementação do interesse. O valor obtido com a aplicação da métrica LOCCConcernPDC é igual a 0 (zero) conseqüentemente, o valor obtido com a aplicação da métrica LOCCConcernPDC dividido pelo valor da métrica TLOCCConcern e pelo valor da métrica LOCCConcernPDC dividido pelo valor da métrica LOCCConcern também são iguais a 0 (zero), o que indica que não há linhas de código do interesse em componentes parcialmente dedicados. Isso leva a conclusão que referente à linhas de código do interesse de persistência no software Ábaco-OA a separação de interesse é total.

No software Ábaco-OO o valor obtido com a aplicação da métrica TLOCCConcern é igual a 2330 (dois mil trezentos e trinta) e o valor obtido com a aplicação da métrica LOCCConcern é 2273 (dois mil duzentos e setenta e três), a divisão do valor da métrica LOCCConcern pelo valor da métrica TLOCCConcern é igual a 0.97 (zero ponto noventa e sete), o que indica que 97% (noventa e sete por cento) das linhas de código estão em componentes totalmente dedicados. O valor obtido com a aplicação da métrica LOCCConcernPDC é igual a 57 (cinquenta e sete), e a divisão do valor da métrica LOCCConcernPDC pelo valor métrica TLOCCConcern é igual a 0.03 (zero ponto zero três), o que indica que 3% (três por cento) do total de atributos do interesse avaliado estão “espalhados” no código base. Para esta aplicação o valor obtido com a aplicação da métrica LOCCConcernPDC dividido pelo valor da métrica LOCCConcern é igual 0.03 (zero ponto zero três), o que indica que as linhas de código do interesse em componentes parcialmente dedicados representam 3% (três por cento) das linhas componentes totalmente dedicados. Isso leva a conclusão que referente à persistência no software Ábaco-OO os atributos do interesse estão 3% (três por cento) “espalhadas”.

No software CDStore-OA o valor obtido com a aplicação da métrica TLOCCConcern é igual a 2409 (dois mil quatrocentos e nove) e o valor obtido com a aplicação da métrica LOCCConcern também é igual a 2409 (dois mil quatrocentos e nove), o valor do métrica

LOCCConcern dividido pelo valor métrica TLOCCConcern é igual a 1 (um), o que indica que 100% (cem por cento) das linhas de código estão em componentes totalmente dedicados à implementação do interesse. O valor obtido com a aplicação da métrica LOCCConcernPDC é igual a 0 (zero) conseqüentemente, o valor da métrica LOCCConcernPDC dividido pelo valor da métrica TLOCCConcern e pelo valor da métrica LOCCConcern dividido pelo valor da métrica LOCCConcern também são iguais a 0 (zero), o que indica que não há linhas de código do interesse em componentes parcialmente dedicados. Isso leva a conclusão que referente à linhas de código do interesse de persistência no software CDStore-OA a separação de interesse é total.

No software CDStore-OO o valor obtido com a aplicação da métrica TLOCCConcern é igual a 2319 (dois mil trezentos e dezenove) e o valor obtido com a aplicação da métrica LOCCConcern é 2268 (dois mil duzentos sessenta e oito) e , a divisão do valor da métrica LOCCConcern pelo valor da métrica TLOCCConcern é igual a 0.98 (zero ponto noventa e oito), o que indica que 98% (noventa e oito por cento) das linhas de código estão em componentes totalmente dedicados. O valor obtido com a aplicação da métrica LOCCConcernPDC é igual a 51 (cinquenta e um), e a divisão do valor obtido com a aplicação da métrica LOCCConcernPDC pelo valor da métrica TLOCCConcern é igual a 0.02 (zero ponto zero dois), o que indica que 2% (dois por cento) do total de linhas de código do interesse avaliado estão “espalhados” no código base. Para esta aplicação o valor obtido com a aplicação da métrica TLOCCConcernPDC dividido pelo valor da métrica LOCCConcern é igual a 0.02 (zero ponto zero dois), o que indica que as linhas de código dedicadas ao interesse de componentes parcialmente dedicados representam 2% (dois por cento) das linhas de código dos componentes totalmente dedicados. Isso leva a conclusão que referente à persistência no software CDStore-OO as linhas de código do interesse estão 2% (dois por cento) “espalhadas”.

Pelas definições das regras heurísticas R07 e R08, é possível concluir que nas duas aplicações OO, os resultados obtidos demonstram um entrelaçamento das linhas de código, do interesse, que na versão OA não ocorre.

**Tabela 25 - Aplicação das regras Heurísticas Comparativas R07 e R08**

Regras Heurísticas Comparativas R07 e R08						
Software	TLOC Concern	LOC Concern	LOCCConcern/ TLOCCConcern	LOCCConcern PDC	LOCCConcern PDC/ TLOCCConcern	LOCCConcern PDC/ LOCCConcern
Ábaco_OA	2409	2409	1	0	0	0
Ábaco_OO	2330	2273	0,97	57	0,03	0,03
CDSStore_OA	2409	2409	1	0	0	0
CDSStore_OO	2319	2268	0,98	51	0,02	0,02

## CONCLUSÃO

Com a aplicação das métricas e regras heurísticas foi possível verificar uma diferença entre os resultados obtidos com as métricas tradicionais, as métricas de separação de interesse e as métricas propostas neste trabalho. Com a aplicação das métricas tradicionais, é possível identificar um aumento de complexidade do código. Nos softwares orientados a aspectos, isso foi verificado tanto no software CDStore quanto no software Ábaco, com aplicação das métricas VS (tamanho do vocabulário), NOA (Número de atributos), NOO (Número de operações), WOC (Peso de operação por componente) e LOC (Linhas de código). Para todas essas métricas o valor encontrado nas versões OA dos *softwares* analisados foi maior que o valor encontrado para a versão OO. Pela definição dessas métricas podemos concluir que na versão OA houve um pequeno aumento do: número de componentes, número de atributos, número de operações, peso das operações por componente, no número de linhas de código. Este aumento torna os softwares OA mais difícil de entender, e mais difícil de encontrar os locais que devem ser alterados.

Dentre as métricas tradicionais apenas DIT (Profundidade da árvore de herança), apresentou resultados diferentes entre as aplicações, não sendo possível assim determinar se a complexidade provocada pela profundidade da árvore de herança é melhorada no paradigma orientado a aspectos com os dois aplicativos utilizados.

Embora tenha sido constatado esse aumento da complexidade nas métricas tradicionais, com as métricas de separação de interesse e com a aplicação das regras heurísticas e métricas propostas é possível identificar uma melhora significativa na separação do interesse. Pela definição das métricas: CDC (difusão do interesse em componentes), CDO (difusão do interesse em operações), CDPDO (Difusão do interesse em operações parcialmente dedicadas), CDPDC (Difusão do interesse em componentes parcialmente dedicados), NOAConcernPDC (Número de atributos do interesse em componentes parcialmente dedicados), LOCCConcernPDC (Número de linhas de código do interesse em componentes parcialmente dedicados), pode-se concluir que, na versão OA houve uma melhora significativa na separação do interesse dos: componentes, das operações, dos atributos e linhas de código. Na versão OA não há espalhamento e em entrelaçamento do interesse com o código base. Pode-se concluir que o interesse está totalmente separado do código base, isso os torna mais fácil de entender, proporciona uma melhor reusabilidade e manutenibilidade.

Na versão OA há um aumento do número de componentes, operações, atributos e linhas de código, comparados com a versão OO. Porém esse aumento é compensado por uma melhora na separação do interesse.

## Projetos Futuros

Implementação de uma ferramenta para contar as métricas propostas;

Estudo de caso utilizando as métricas e regras heurísticas propostas em *softwares* com mais de um interesse transversal;

Propostas para novas métricas e regras heurísticas, para comparação entre softwares OO e OA;

## REFERÊNCIAS BIBLIOGRÁFICAS

- ASPECTWERKZ WEBSITE. Simple, Dynamic, Lightweight and Powerful AOP for Java.  
<http://aspectwerkz.codehaus.org/>
- ELRAD, T., FILMAN R., BADER A. Aspect-Oriented Programming. *Communications of the ACM*, vol 44, pp 29-32, 2001.
- GRISWOLD, W.G., SHONLE, M., SULLIVAN, K., SONG, Y., CAI, Y., RAJAN, H.  
*Modular Software Design with Crosscutting Interfaces*, IEEE Software, pp 51-60, 2006.
- MEZINI, M., OSTERMANN, K. Variability Management with Feature Oriented Programming and Aspects. In: SIGSOFT/FSE, ACM, 2004.
- KICZALES, G., LAMPING, J., MENDHEKAR, A., MAEDA, C., LOPES, C., LOINGTIER, J., IRVING, J. Aspect Oriented Programming. In: *Proceedings of 11 ECOOP*. pp. 220-242, 1997.
- TARR, P., OSSHER, H., SUTTON, S. Hyper/JTM : Multi-dimensional Separation of Concerns for Java. In: *Proc. of the 24th International Conference on Software Engineering*. Orlando, Florida, 2002.
- FIGUEIREDO, E. M. L. : Uma Abordagem Quantitativa para Desenvolvimento de Software Orientado a Aspectos. Dissertação de Mestrado, Departamento de informática, PUC-Rio Rio de Janeiro, 2006.
- FIGUEIREDO, E. M. L.; STAA, A. : Avaliação de um Modelo de Qualidade para Implementação Orientada a Objetos e Orientada a Aspectos. Monografia em Ciência da Computação, Departamento de informática, PUC-Rio Rio de Janeiro, 2005.
- SANT'ANNA, C. N. : Manutenibilidade e Reusabilidade de Software Orientado a Aspectos: Um Framework de Avaliação. Dissertação de Mestrado, Departamento de informática, PUC-Rio Rio de Janeiro, 2004.
- CAMARGO, V. V. : FRAMEWORKS transversais: definições, classificações, arquitetura e um processo de desenvolvimento de software. Tese de Doutor em Ciências - Ciências de Computação e Matemática Computacional, UPS São Carlos, 2006.