

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL  
FACULDADE DE INFORMÁTICA

VANESSA BARBISAN PIRES

**ANÁLISE DE IMPACTO DE MUDANÇA DE SOFTWARE:  
UMA METODOLOGIA BASEADA EM ONTOLOGIAS**

Porto Alegre  
2006

**VANESSA BARBISAN PIRES**

**ANÁLISE DE IMPACTO DE MUDANÇA DE SOFTWARE:  
UMA METODOLOGIA BASEADA EM ONTOLOGIAS**

Trabalho apresentado ao Programa de Pós-Graduação em  
Ciência da Computação como requisito parcial à obtenção  
do grau de Mestre em Ciência da Computação.

Pontifícia Universidade Católica do Rio Grande do Sul

Orientador: Prof. D. Sc. Marcelo Blois Ribeiro.

Porto Alegre, 2006

### Dados Internacionais de Catalogação na Publicação (CIP)

P667a Pires, Vanessa Barbisan  
Análise de impacto da mudança de software : uma metodologia baseada em ontologias / Vanessa Barbisan Pires. – Porto Alegre, 2008.  
141 f.

Diss. (Mestrado) – Fac. de Informática, PUCRS.  
Orientador: Prof. Dr. Marcelo Blois Ribeiro.

1. Informática. 2. Engenharia de Software. 4. Ontologia.  
5. Software – Manutenção. I. Ribeiro, Marcelo Blois. II. Título.

CDD 005.1

Ficha Catalográfica elaborada pelo  
Setor de Tratamento da Informação da BC-PUCRS



Pontifícia Universidade Católica do Rio Grande do Sul  
FACULDADE DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

## TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "**Análise de Impacto de Mudança de Software: Uma Metodologia Baseada em Ontologias**", apresentada por Vanessa Barbisan Pires, como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Sistemas de Informação, aprovada em 23/01/2007 pela Comissão Examinadora:



---

Prof. Dr. Marcelo Blois Ribeiro -  
Orientador (a)

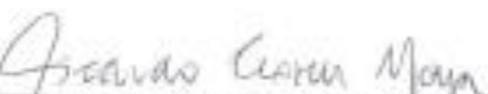
PPGCC/PUCRS



---

Prof. Dr. Ricardo Melo Bastos -

PPGCC/PUCRS



---

Prof. Dr. Ricardo Choren Noya -

IME/RJ

Homologada em...../...../....., conforme Ata No. .... pela Comissão Coordenadora.

---

Prof. Dr. Fernando Luís Dotti  
Coordenador.

PUCRS

Campus Central

Av. Ipiranga, 6681 - P. 16 - sala 106 - CEP: 91519-900

Fone: (51) 3320-3611 - Fax (51) 3320-3621

E-mail: [ppgcc@inf.pucrs.br](mailto:ppgcc@inf.pucrs.br)

[www.pucrs.br/facinf/pos](http://www.pucrs.br/facinf/pos)

*Dedico essa dissertação ao meu namorado José Olavo. Aos meus amigos, além de mãe e padrasto, Rose e Jaurês. A meu pai, que torce mesmo distante. Às minhas irmãs Michele, Fran, Carol e Sthefane. E à minha tia Jane, minha vó Ruth e meu vô Ruy.*

## AGRADECIMENTOS

*Ao meu orientador, Prof. Dr. Marcelo Blois Ribeiro, pelo incentivo, pela perseverança nos momentos difíceis e pelo otimismo de que o caminho estava certo. Agradeço principalmente por ter aceitado o desafio de orientar uma aluna que não podia dedicar integralmente seu tempo para desenvolver a pesquisa.*

*Aos professores Dr. Ricardo Bastos e Dr. Toacy Oliveira, que deram suas contribuições ao longo do desenvolvimento desse trabalho, revisando e criticando os volumes produzidos.*

*Aos professores Dr<sup>a</sup> Márcia Campos, minha orientadora no trabalho de conclusão do curso de graduação em Ciência da Computação, e Dr. Michael Móra, que teve paciência de revisar os mais diversos assuntos para a prova do POSCOMP, agradeço pelas cartas de recomendação.*

*Ao professor Dr. Avelino Zorzo que, na condição de coordenador do Programa de Pós-Graduação, me concedeu uma bolsa de estudos, possibilitando o término dessa pesquisa.*

*Ao meu chefe Eduardo Peres que, mais do que ter me incentivado sugeriu a linha de pesquisa a ser abordada. Agradeço a compreensão e preocupação ao longo desse período e posso dizer que foi mais do que meu diretor, foi meu amigo. E à DBServer agradeço por ter apostado nesta minha qualificação.*

*Ao meu hoje amigo Arthur Sfreddo, que na condição de meu chefe na PROCERGS, soube apoiar o meu projeto e acreditar na minha capacidade profissional, mesmo sem nunca antes ter trabalhado comigo, flexibilizando meus horários para que eu pudesse assistir às aulas.*

*Aos colegas do Grupo de Sistemas de Software Inteligentes, que contribuíram com suas opiniões e idéias. Agradeço ao colega Rodrigo Noll pelo nosso trabalho em equipe durante o desenvolvimento dessa pesquisa. Aos colegas Oscar e Ana Paula no apoio ao desenvolvimento da ferramenta. Também a todos os alunos do Programa de Pós-Graduação em Ciência da Computação que participaram dos experimentos propostos.*

*A todos os meus amigos que estiveram presentes nesse momento, torcendo, me incentivando e ajudando das mais diversas maneiras. Principalmente, ao amigo Ricardo Angrisani, que junto comigo acreditou ser possível concluir o mestrado. Ao meu quase irmão Tasso que certamente foi a grande inspiração para eu também querer o título de mestre! Á Silvana e ao Maurício que quase diariamente me escutaram e deram força. Á Ângela, Carol, Tanara, André, Cássio, Ed, Eduardo, Pescoço e Marcus que ouviram muitos: “Não posso ir”. E a minha cunhada e amiga Janise que se dispôs a revisar esse trabalho.*

*À minha família agradeço pelo apoio incondicional. Ao tio e a mãe por todo carinho e incentivo, segurando minha onda nos momentos difíceis. Ao meu pai, pela torcida. As minhas irmãs, pela presença. A minha tia, que diz que algo está a minha espera após a conclusão desse trabalho. E ao orgulho que meus avós tem pela neta.*

*E, especialmente, ao meu amigo além de namorado, Olavo, pela ajuda nos trabalhos, por compreender as ausências e pela paz e segurança que me transmitiu nos meus momentos de dúvida.*

*"O mais importante para o homem é crer em si mesmo.  
Sem esta confiança em seus recursos, em sua inteligência,  
em sua energia, ninguém alcança o triunfo a que aspira"  
Thomas Atkinson.*

## RESUMO

Quando um requisito de mudança é solicitado durante um projeto de manutenção, toda a informação referente a ele deve ser rastreada a fim de que seja possível desenvolvê-lo. Nessa realidade está inserida a análise de impacto de uma mudança que visa, entre outras coisas, estabelecer a probabilidade de impacto que um artefato terá. O principal objetivo desse trabalho é o desenvolvimento de uma metodologia de análise de impacto de mudança em projetos de manutenção de software. Para que seja possível realizá-la, é preciso rastrear os artefatos que compõem o sistema. O rastreamento é realizado pelos conceitos da ontologia gerada durante o processo de desenvolvimento de software a partir do modelo de domínio.

**Palavras-chave:** Projetos de manutenção de software, análise de impacto, rastreabilidade, ontologia, processo de desenvolvimento de software.

## **ABSTRACT**

When a change requirement is generated during a software project, all information necessary for its development must be traced. In this context, the change impact analysis focus on finding the probability of an artifact impact to be modified. The main goal of this work is to develop a change impact methodology for software maintenance projects. To analyze the change impact it is necessary to use a traceability methodology. In our approach, the traceability is enforced by the ontology concepts that are generated from the domain model during the software development process. The methodology validation will be made by an experiment.

**Keywords:** software maintenance project, impact analysis, traceability, ontology, software development process.

## LISTA DE FIGURAS

Figura 1 – Exemplo de ontologia. ....	20
Figura 2 – Fluxo para determinar análise de impacto de mudança. ....	27
Figura 3 – Processo de análise de impacto proposto por Bohner. ....	27
Figura 4 – Grafo com relacionamento entre objetos do ciclo de vida do software (OCV) .....	29
Figura 5 – Modelo de rastreamento entre os artefatos do sistema.....	31
Figura 6 – Relacionamento entre mudanças, artefatos e grau de influência entre eles. ....	32
Figura 7 – Relacionamento entre artefatos e requisitos de mudança. ....	33
Figura 8 – Integração da disciplina de Engenharia Ontológica ao RUP .....	39
Figura 9 – Representação da rastreabilidade semântica através de ontologia.....	40
Figura 10 – Exemplo de diagrama de classes.....	47
Figura 11 – Aplicação do método Fast&&Serious.....	48
Figura 12 – Requisito de mudança A, no domínio de livros. ....	48
Figura 13 – Exemplo de diagrama de classes após alteração do requisito de mudança A. ....	48
Figura 14 – Níveis de influência entre artefatos na metodologia TIAM. ....	49
Figura 15 - Valores de influência normalizados entre artefatos na metodologia TIAM. ....	50
Figura 16 - Relacionamento entre conceitos da ontologia e artefatos de software. ....	50
Figura 17 - Influência do conceito no artefato ( $I_c$ ) do conceito Autor.....	52
Figura 18 – Exemplo em OWL da restrição <i>allValuesFrom</i> .....	55
Figura 19 – Sub ontologia equivalente ao domínio de livros.....	56
Figura 20 – Requisito de mudança B no domínio de livros. ....	56
Figura 21 – Sub ontologia equivalente ao requisito de mudança no domínio de livros.....	56
Figura 22 – Parte de um modelo de domínio de livros.....	57
Figura 23 – Ontologia equivalente a parte de um modelo de domínio de livros.....	58
Figura 24 – Ontologia equivalente ao requisito de mudança A no domínio de livros. ....	58
Figura 25 – Requisito de mudança C no domínio de livros. ....	59
Figura 26 – Ontologia equivalente ao requisito de mudança C no domínio de livros. ....	59
Figura 27 – Exemplo de radicalização de palavras. ....	61
Figura 28 – Sub ontologia equivalente ao domínio de livros, destacando a propriedade <i>objectProperty</i> . ....	62
Figura 29 – Exemplo de tesouro.....	63
Figura 30 – Exemplo de cálculo do Impacto acumulado ( $Ia_{(x)}$ ). ....	65
Figura 31 – Exemplo do cálculo do Impacto no Artefato pelo Relacionamento ( $Iar_{(x,y)}$ ) .....	66
Figura 32 – Exemplo de cálculo da Propagação do impacto ( $Pi$ ). ....	66
Figura 33 – Passos da metodologia de análise de impacto baseada em ontologias.....	67
Figura 34 – Desenvolvimento do modelo conceitual do requisito de mudança.....	68
Figura 35 – Gerar ontologia do requisito de mudança a partir de um modelo conceitual.....	68
Figura 36 – Intersecção entre ontologias .....	69

Figura 37 – Identificação dos conceitos da ontologia do requisito de mudança na ontologia do sistema .....	69
Figura 38 – Identificação de inclusão de um conceito do tipo <i>dataTypeProperty</i> .....	70
Figura 39 – Identificação de alteração de um conceito do tipo <i>dataTypeProperty</i> .....	70
Figura 40 – Identificação de um conceito do tipo <i>dataTypeProperty</i> que não sofre mudança	70
Figura 41 – Identificação de relacionamento entre conceitos .....	71
Figura 42 – Rastreabilidade de relação conceito-artefato .....	72
Figura 43 – Rastreabilidade de alteração de um conceito do tipo <i>dataTypeProperty</i> .....	73
Figura 44 – Cálculos da relação <i>dataTypeProperty</i> -Artefato .....	73
Figura 45 – Rastreabilidade de inclusão de um conceito do tipo <i>dataTypeProperty</i> .....	74
Figura 46 – Cálculos da relação estrutural <i>class</i> -Artefato .....	75
Figura 47 – Alteração de regra de negócio associada a conceito .....	75
Figura 48 – Cálculos da relação semântica <i>class</i> -Artefato .....	75
Figura 49 – Identificação de um conceito do tipo <i>dataTypeProperty</i> que não sofre mudança	76
Figura 50 – Cálculos da relação <i>class</i> -Artefato sem alterações. ....	76
Figura 51 – Rastreabilidade de relação conceito-conceito .....	77
Figura 52 – Rastreabilidade de artefatos na relação conceito-conceito .....	77
Figura 53 – Cálculo do Impacto acumulado ( $Ia_{(x)}$ ) .....	78
Figura 54 – Tela com os conceitos do tipo <i>class</i> e <i>dataTypeProperty</i> .....	80
Figura 55 – Entrada das influências do conceito e tipo de diagrama .....	81
Figura 56 – Tela com os conceitos do tipo <i>objectProperty</i> .....	81
Figura 57 – Tela para informação se houve alteração de regra de negócio .....	82
Figura 58 – Comparação de ontologias através de código XML .....	83
Figura 59 – Tela com menu de escolha para comparação da ontologias e cálculo do impacto	83
Figura 60 – Requisito de mudança para o sistema <i>IsGym</i> .....	84
Figura 61 – Dicionário de Dados <i>IsGym</i> .....	85
Figura 62 – Modelo de conceitual do requisito de mudança .....	85
Figura 63 – Ontologia do requisito de mudança representada em XML .....	86
Figura 64 – Ontologia do requisito de mudança representada graficamente. ....	87
Figura 65 – Representação parcial da ontologia do sistema, com conceitos do requisito de mudança assinalados .....	88
Figura 66 – Identificação do conceito “meta”. ....	88
Figura 67 – Identificação do conceito “FichaBiometrica”. ....	88
Figura 68 – Identificação do conceito “Mensalidade”, com alteração estrutural .....	89
Figura 69 – Identificação do conceito “Mensalidade”, com alteração de regra de negócio .....	89
Figura 70 – Rastreamento dos artefatos ligados ao conceito “meta”. ....	90
Figura 71 – Cálculo dos artefatos ligados ao conceito “meta”. ....	90
Figura 72 – Rastreamento dos artefatos ligados ao conceito “FichaBiometrica”. ....	90
Figura 73 – Cálculos dos artefatos ligados ao conceito “FichaBiometrica”. ....	91
Figura 74 – Rastreamento dos artefatos ligados ao conceito “Mensalidade” .....	91
Figura 75 – Cálculos dos artefatos ligados ao conceito “Mensalidade” - estrutural. ....	91
Figura 76 – Cálculos dos artefatos ligados ao conceito “Mensalidade” – regra de negócio. ....	91
Figura 77 – Rastreamento dos artefatos ligados ao conceito “Cliente”. ....	92
Figura 78 – Cálculos dos artefatos ligados ao conceito “Cliente”. ....	93
Figura 79 – Rastreamento dos artefatos ligados ao conceito “Funcionário” .....	94
Figura 80 – Cálculos dos artefatos ligados ao conceito “Funcionário” .....	95
Figura 81 – Cálculo da média ponderada. ....	97
Figura 82 – Variáveis independentes e dependentes do estudo experimental .....	104
Figura 83 – Gráfico de barras relativo à precisão dos artefatos. ....	112
Figura 84 – Gráfico de dispersão para a variável precisão .....	114

Figura 85 – Modelo de caso de uso de negócio para o contexto Criar Ficha de Acompanhamento de Cliente. ....	127
Figura 86 – Modelo de atividades de negócio para o contexto Criar Ficha de Acompanhamento de Cliente. ....	128
Figura 87 – Modelo de interação de negócio para o contexto Criar Ficha de Acompanhamento de Cliente. ....	128
Figura 88 – Modelo de objetos de negócio para o contexto Criar Ficha de Acompanhamento de Cliente. ....	128
Figura 89 – Diagrama de casos de uso do sistema <i>IsGym</i> . ....	129
Figura 90 – Descrição do caso de uso Criar Ficha Pessoal. ....	130
Figura 91 – Diagrama de atividades Criar Ficha Pessoal. ....	130
Figura 92 – Modelo de classe de análise Criar Ficha Pessoal. ....	131
Figura 93 – Modelo de colaboração em análise Criar Ficha Pessoal. ....	131
Figura 94 – Modelo de sequencia Criar Ficha Pessoal. ....	131
Figura 95 – Diagrama de classes do sistema <i>IsGym</i> . ....	132
Figura 96 – Resultado do questionário aplicado para conhecer o perfil dos alunos. ....	133
Figura 97 – Exemplo de preenchimento do relatório de conjunto de impacto. ....	134
Figura 98 – Requisito de mudança 2 para sistema <i>IsGym</i> . ....	135
Figura 99 – Requisito de mudança 3 para sistema <i>IsGym</i> . ....	136
Figura 100 – Artefatos impactados pelo requisito de mudança 1. ....	137
Figura 101 – Artefatos impactados pelo requisito de mudança 2. ....	137
Figura 102 – Artefatos impactados pelo requisito de mudança 3. ....	138
Figura 103 – Tutorial fornecido para os participantes executarem a análise de impacto ontológica. ....	139
Figura 104 – Planilha para preenchimento das variáveis da metodologia de análise de impacto ontológica. ....	140
Figura 105 – Tutorial fornecido para os participantes executarem a análise de impacto por requisitos. ....	140
Figura 106 – Planilha para preenchimento das variáveis da metodologia de análise de impacto por requisitos. ....	141
Figura 107 – Questionário fornecido para os participantes. ....	141

## LISTA DE TABELAS

Tabela I – Matriz de conectividade .....	29
Tabela II – Atividades por fase da Engenharia Ontológica.....	38
Tabela III – Passos do método Fast&&Serious.....	45
Tabela IV – Passos dos atributos no método Fast&&Serious .....	45
Tabela V – <i>Guideline</i> para classificação da Influência do conceito no artefato ( <i>Ic</i> ).....	51
Tabela VI – Classificação do tipo de diagrama .....	53
Tabela VII – Probabilidade de impacto dos artefatos rastreados. ....	96
Tabela VIII – Tabela de contingência. ....	106
Tabela IX – Escalas das variáveis. ....	111
Tabela X – Tabulação dos valores brutos obtidos após a execução do experimento.....	112
Tabela XI – Teste de normalidade <i>Shapiro-Wilk</i> para a variável <i>precisão</i> . ....	114
Tabela XII – Teste de <i>Levene</i> para igualdade das variâncias sobre a variável <i>precisão</i> . ....	115
Tabela XIII – Teste não paramétrico de <i>Mann-Whitney</i> para a variável <i>precisão</i> . ....	115
Tabela XIV – Estatística descritiva para a variável <i>precisão</i> . ....	116
Tabela XV – Resultados da avaliação qualitativa. ....	117
Tabela XVI – Média da satisfação das questões sobre as abordagens. ....	117

## LISTA DE FÓRMULAS

Fórmula 1 – Artefatos que serão alterados .....	28
Fórmula 2 – Cálculo do número de Pontos por Estado .....	45
Fórmula 3 – Cálculo da Complexidade do Método.....	46
Fórmula 4 – Cálculo dos Pontos por Comportamento .....	46
Fórmula 5 – Cálculo dos Pontos por Classe .....	46
Fórmula 6 – Cálculo do peso do artefato.....	47
Fórmula 7 – Normalização dos elos na metodologia TIAM .....	49
Fórmula 8 – Cálculo do Impacto no artefato pela mudança ( $Iam_{(x,y)}$ ) .....	54
Fórmula 9 – Cálculo do Impacto acumulado ( $Ia_{(x)}$ ).....	65
Fórmula 10 – Cálculo da Propagação do impacto ( $Pi$ ).....	65
Fórmula 11 – Cálculo do Impacto no artefato pelo relacionamento ( $Iar_{(x,y)}$ ).....	66
Fórmula 12 – Cálculo do Percentual de probabilidade de impacto do artefato ( $Ppi_{(y)}$ ). .....	79
Fórmula 13 – Cálculo da média ponderada. ....	79
Fórmula 14 – Cálculo da variável Precisão. ....	101

## LISTA DE ABREVIATURAS

GQM	<i>Goal, Question, Metric</i>
JSP	<i>JavaServer Pages</i>
OO	<i>Orientação a Objetos</i>
OWL	<i>Web Ontology Language</i>
PLN	<i>Processamento de Linguagem Natural</i>
RUP	<i>Rational Unified Process</i>
SLOC	<i>Source Line of Code</i>
TIAM	<i>Trace-based Impact Analysis Methodology</i>
UML	<i>Unified Modeling Language</i>
XML	<i>eXtensible Markup Language</i>
WMSWM	<i>Workshop de Manutenção de Software Moderna</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>17</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>19</b>
2.1	ONTOLOGIA	19
2.2	ONTOLOGIA E PROCESSO DE DESENVOLVIMENTO DE SOFTWARE	21
2.3	PROJETO DE MANUTENÇÃO DE SOFTWARE	22
2.4	RASTREABILIDADE	23
2.5	ANÁLISE DE IMPACTO	25
2.5.1	<i>Modelos de Análise de Impacto</i>	26
2.5.1.1	Modelos de Bohner	27
2.5.1.2	Metodologia baseada em rastreabilidade	30
2.6	ENGENHARIA DE SOFTWARE EXPERIMENTAL	34
<b>3</b>	<b>MODELANDO A ANÁLISE DE IMPACTO DE MUDANÇA</b>	<b>37</b>
3.1	ENGENHARIA ONTOLÓGICA	38
3.2	RASTREABILIDADE SEMÂNTICA ATRAVÉS DE ONTOLOGIA	40
3.3	INFLUÊNCIA DA COMPLEXIDADE DE ARTEFATO	41
3.3.1	<i>Complexidade de Artefato</i>	42
3.3.1.1	Métricas de produto de software	42
3.3.1.2	Métricas para UML	44
3.3.1.2.1	Complexidade do Diagrama de Classes	44
3.3.2	<i>Uso da Complexidade de Artefato</i>	46
3.4	INFLUÊNCIA DO CONCEITO DA ONTOLOGIA NOS ARTEFATOS	49
3.5	INFLUÊNCIA DO TIPO DE DIAGRAMA NA ANÁLISE DE IMPACTO	52
3.6	INFLUÊNCIA DO TIPO DE MUDANÇA NA ANÁLISE DE IMPACTO	53
3.6.1	<i>Tipo de Mudança de Regras de Negócio</i>	54
3.6.2	<i>Tipo de Mudança Estrutural</i>	55
3.6.2.1	Alteração de um conceito da ontologia	55
3.6.2.2	Inclusão de um conceito na ontologia	57
3.6.2.2.1	Inclusão de conceito do tipo <i>dataTypeProperty</i>	57
3.6.2.2.2	Inclusão de conceito do tipo <i>class</i>	59
3.7	INFLUÊNCIA DO RELACIONAMENTO ENTRE CONCEITOS	60
3.7.1	<i>Processamento de Linguagem Natural</i>	60
3.7.1.1	Processamento Morfossintático	61
3.7.1.2	Processamento Semântico	62
3.7.2	<i>Cálculo da Influência do Relacionamento entre Conceitos</i>	64
<b>4</b>	<b>METODOLOGIA PARA ANÁLISE DE IMPACTO DE MUDANÇA</b>	<b>67</b>
4.1	GERAR ONTOLOGIA DO REQUISITO DE MUDANÇA	68
4.2	IDENTIFICAR CONCEITOS NA ONTOLOGIA DO SISTEMA	69
4.3	RASTREAR ARTEFATOS	71
4.3.1	<i>Relação Conceito-Artefato</i>	72
4.3.1.1	Relação <i>dataTypeProperty</i> -Artefato	73
4.3.1.2	Relação <i>class</i> -Artefato	74
4.3.2	<i>Relação Conceito-Conceito</i>	76
4.4	LISTAR ARTEFATOS	79
4.5	FERRAMENTA	80
4.6	EXEMPLO DE APLICAÇÃO DA METODOLOGIA	83

4.6.1	<i>Gerar Ontologia do Requisito de Mudança</i> .....	84
4.6.2	<i>Identificar Conceitos na Ontologia do Sistema</i> .....	87
4.6.3	<i>Rastrear Artefatos</i> .....	89
4.6.4	<i>Listar e Calcular a Probabilidade dos Artefatos</i> .....	96
<b>5</b>	<b>VALIDAÇÃO DA METODOLOGIA</b> .....	<b>99</b>
5.1	DEFINIÇÃO.....	99
5.1.1	<i>Objetivo Global</i> .....	100
5.1.2	<i>Objetivo do Estudo</i> .....	100
5.1.3	<i>Objetivo da Medição</i> .....	100
5.1.4	<i>Questão</i> .....	101
5.1.5	<i>Métricas</i> .....	101
5.2	PLANEJAMENTO.....	102
5.2.1	<i>Seleção do Contexto</i> .....	102
5.2.2	<i>Formulação das Hipóteses</i> .....	103
5.2.3	<i>Seleção das Variáveis</i> .....	104
5.2.3.1	Variável Independente.....	104
5.2.3.2	Variável Dependente.....	104
5.2.4	<i>Seleção dos Indivíduos</i> .....	105
5.2.5	<i>Projeto do Experimento</i> .....	105
5.2.5.1	Princípios Genéricos de Projetos.....	105
5.2.5.2	Padrão para Tipo de Projeto.....	106
5.2.6	<i>Instrumentação</i> .....	107
5.2.7	<i>Análise da Validade</i> .....	107
5.2.7.1	Validade Interna.....	108
5.2.7.2	Validade Externa.....	108
5.2.7.3	Validade de Construção.....	108
5.2.7.4	Validade de Conclusão.....	109
5.3	EXECUÇÃO.....	110
5.3.1	<i>Preparação</i> .....	110
5.3.2	<i>Execução</i> .....	111
5.4	ANÁLISE E INTERPRETAÇÃO.....	111
5.4.1	<i>Análise Tabular e Gráfica</i> .....	112
5.4.2	<i>Estatística Descritiva</i> .....	113
5.4.3	<i>Hipótese: Precisão</i> .....	113
5.5	AVALIAÇÃO QUALITATIVA.....	116
5.6	CONSIDERAÇÕES.....	117
<b>6</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS</b> .....	<b>119</b>
	<b>REFERÊNCIAS</b> .....	<b>123</b>
	<b>APÊNDICE A – RESULTADOS DO EXPERIMENTO</b> .....	<b>127</b>
	<b>APÊNDICE B – INSTRUMENTAÇÃO DO EXPERIMENTO DE VALIDAÇÃO</b> .....	<b>139</b>



## 1 INTRODUÇÃO

A fase inicial do ciclo de vida de um produto de software é o processo de desenvolvimento, aonde irá se construir o sistema desejado pelo cliente. Quando esta é finalizada, inicia-se a manutenção do produto que, segundo a organização do *Workshop* de Manutenção de Software Moderna (WMSWM)[WMS04], é a fase do ciclo de vida do software que consome mais recursos, sendo responsável por aproximadamente 90% do custo total deste. Isso se dá por vários motivos, dentre eles mudanças tecnológicas, alteração de necessidades do usuário e falhas encontradas após sua instalação.

Para tentar manter a qualidade do produto e conseguir estimar o esforço e tempo necessários para realizar uma mudança no software, é preciso que se tenha uma maneira otimizada para descobrir todas as partes do sistema que serão afetadas pela alteração. Essas podem ser obtidas por um modelo de análise de impacto, que irá determinar de que forma os artefatos são impactados e qual a probabilidade disso ocorrer. Existem diversas pesquisas nessa área, com propostas que visam encontrar o conjunto de artefatos impactados mais próximo da realidade, ou seja, encontrar com precisão os elementos que serão efetivamente modificados.

Para que seja possível obter esse conjunto, é necessário possuir um mecanismo de rastreabilidade que suporte a análise de impacto. Um autor que se destaca em pesquisas nesse sentido é Bohner[BOH96] com diversos artigos e livro publicados. Ele propôs uma metodologia de análise de impacto, rastreando os artefatos através das ligações entre eles, representadas em um grafo de conectividade. Outro trabalho interessante foi realizado por O'Neal[ONE03], onde a probabilidade de impacto nos artefatos é determinada e a rastreabilidade é feita através dos requisitos do sistema.

Outro método utilizado para realizar o rastreamento de artefatos faz uso de mecanismos semânticos para tal, visando uma maior assertividade para encontrar o conjunto de elementos que serão alterados. Através de pesquisa realizada por Noll[NOL05a], notou-se que ontologias poderiam ser integradas ao processo de desenvolvimento de software e, através de seus conceitos, realizar a rastreabilidade semântica dos artefatos. Uma das contribuições de

utilizar ontologia nesse contexto é que esta permite integrar os elementos produzidos, desde a modelagem de negócio, até a análise e projeto.

Considerando-se que o uso desse tipo de rastreabilidade traria ganho para encontrar mais precisamente os artefatos modificados a partir de um requisito de mudança, foi desenvolvida uma metodologia para análise de impacto baseada em ontologias. Essa é responsável por identificar as alterações que ocorrem na ontologia do sistema e assim mensurar a probabilidade de impacto em cada um dos artefatos.

Para a construção dessa proposta, buscou-se compreender o que são, o que representam e como as ontologias se integram ao processo de desenvolvimento de software. Pesquisou-se saber também, como é definida a fase de manutenção de um produto, onde diversos requisitos de mudança são solicitados. Sendo a diferença entre rastreabilidade e análise de impacto bastante tênue, procurou-se estabelecer fortemente as diferenças entre elas. Ainda como parte do referencial teórico, abordado no capítulo 2, estudou-se a disciplina de Engenharia de Software Experimental com a finalidade de utilizá-la como base para o experimento de validação dessa proposta.

Com a intenção de conhecer quais os mecanismos utilizados para realizar a rastreabilidade de artefatos através da ontologia, foram estudadas a Engenharia Ontológica e a rastreabilidade semântica. Com o auxílio desse estudo, presente no capítulo 3, foi possível determinar as influências que uma mudança na ontologia do sistema exerce sobre os artefatos relacionados aos conceitos desta.

Surgiu então, a metodologia que será apresentada no capítulo 4 dessa proposta, estabelecendo os passos necessários para realizá-la, cuja principal contribuição é a de possibilitar maior precisão para realizar a análise de impacto de um requisito de mudança. Para automatizar essa tarefa, foi desenvolvida uma ferramenta onde, uma vez determinada a influência existente entre conceitos, artefatos e tipos de mudança, é possível identificar os elementos com maior probabilidade de serem impactados. O experimento realizado para validação da metodologia é apresentado no capítulo 5, considerando todos os passos necessários para seu desenvolvimento. Por fim, são feitas algumas considerações e sugeridos alguns trabalhos futuros.

## 2 REFERENCIAL TEÓRICO

Esse capítulo descreve o referencial teórico deste trabalho expondo os conhecimentos necessários para entendimento do problema e para a solução. Primeiramente é preciso saber o que as ontologias representam e qual sua integração com o processo de desenvolvimento de software, sendo aqui utilizado o *Rational Unified Process* (RUP). Também é necessário conhecer a fase do ciclo de vida do software caracterizada por projetos de manutenção deste, com relação à rastreabilidade e ao impacto causado por uma requisição de mudança no sistema. A Engenharia de Software Experimental é descrita, já que essa tem como objetivo validar propostas da área de engenharia de software através de experimentos.

### 2.1 ONTOLOGIA

Um processo de construção de ontologia, no que diz respeito à Ciência da Computação, representa, segundo Gruber[GRU93], a aquisição do conhecimento a partir de dados semi-estruturados utilizando um conjunto de métodos, técnicas ou processos automáticos ou semi-automáticos. Esse termo teve origem na computação através da área de Inteligência Artificial, que considera a representação através da ontologia similar a como o conhecimento humano é representado. Segundo Fensel[FEN00], ontologias são desenvolvidas para facilitar o compartilhamento e reuso de informações.

Essa facilidade para compartilhamento de informações foi a principal razão para sua utilização. Afirma-se ainda que a ontologia é mais que um vocabulário padrão e mais do que uma taxonomia, uma vez que assegura que os termos escolhidos são suficientes para especificar e definir conceitos, bem como o relacionamento entre eles, e que inclui a expressão exata do domínio específico do conhecimento.

Gruber[GRU93] definiu ontologia como sendo uma representação formal do conhecimento que é baseada em uma conceituação. Ele afirma que se pode descrever uma ontologia pela definição de um conjunto representacional de termos. Essa definição compõe-se de associações entre nomes de entidades do universo de discurso, formado pelo conjunto de objetos que podem ser representados. Esse conjunto pode ser composto por classes, relacionamentos ou funções, descritos através de um texto com seus significados, e com axiomas formais, responsáveis pela restrição da interpretação e pela boa formação no uso das associações.

A estrutura de dados de uma ontologia é equivalente a um grafo. Cada nodo é um conceito que segundo Geller[GEL04] corresponde a palavras ou frases curtas, sendo tipicamente correspondente a substantivos. Esses conceitos são conectados uns aos outros, sendo o relacionamento mais importante o do tipo É-UM. Alguns nodos possuem outras informações relacionadas, podendo ser atributos, relacionamentos ou regras. A figura 1 abaixo exemplifica uma ontologia em termos de conceitos, relacionamento e regras.

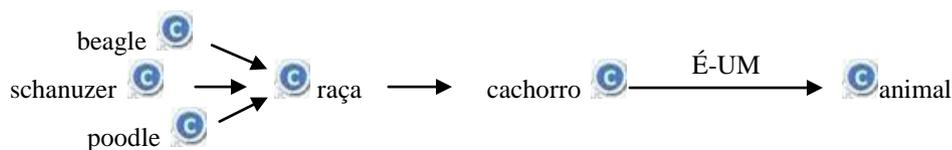


Figura 1 – Exemplo de ontologia.

Existem duas abordagens para construção de ontologias, no ponto de vista da área de Engenharia de Software: uma é a construção de ontologias baseadas nos requisitos do sistema e outra é a construção de uma ontologia para a representação de requisitos. A primeira é vista como subproduto da atividade de Engenharia de Requisitos. Felicíssimo [FEL03] defende esta questão por considerar que é durante a especificação dos requisitos do sistema que se delimita o escopo do projeto bem como se define o domínio da aplicação. Nesta fase, pode-se construir uma ontologia única, formada pela união dos conceitos do problema e da solução, que poderá ser refinada em seguida.

O papel da ontologia, quando utilizada para a representação de requisitos, é de prover a comunicação entre os requisitos através de uma sintaxe e semânticas bem definidas. Como resultado, tem-se um mecanismo baseado no conhecimento para criação e manutenção de documentos de especificação de requisitos do sistema. Para obter-se esse mecanismo, utiliza-se lógica de primeira ordem para definir os componentes da ontologia e identificam-se os axiomas envolvidos com os objetos e a sua interação para responder questões de senso comum.

Tendo em vista que a comunicação entre os requisitos é determinada durante o processo de desenvolvimento de software, na próxima seção a integração de ontologias a este é caracterizada.

## 2.2 ONTOLOGIA E PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

Um processo de desenvolvimento de software pode ser definido, segundo Sommerville[SOM03], como um conjunto de atividades e resultados associados que geram um produto de software. Utilizando o paradigma de orientação a objetos e os conceitos da *Unified Modeling Language* (UML), foi proposta por Jacobson, Booch e Rumbaugh uma metodologia de desenvolvimento de software unificada, o RUP.

Dentre os objetivos do RUP estão o aumento da produtividade e maior qualidade do produto de software produzido. Para que tais metas sejam alcançadas, ele deve ser configurado para cada empresa ou projeto que desejem utilizá-lo. Isso é possível uma vez que o RUP pode ser considerado um *framework* para processos de desenvolvimento de software.

O RUP é dividido em quatro fases distintas, com objetivos específicos: concepção (definição do escopo), elaboração (definição da arquitetura), construção (desenvolvimento) e transição (implantação do produto). Tido como um processo incremental e iterativo, cada uma dessas fases pode suportar uma ou mais iterações. Cada uma dessas iterações possui fluxos específicos de trabalhos definidos pela metodologia, sendo eles: modelagem de negócios, requisitos, análise e projeto, implementação, testes e implantação, que compõem o fluxo de trabalho de engenharia, além da gerência de projetos, gerência de configuração e mudanças e configuração de ambiente, que fazem parte do fluxo de trabalho de suporte.

A integração de ontologias no processo de desenvolvimento de software foi estudada e proposta por Noll[NOL05b]. Em seu trabalho, ontologias são relacionadas ao modelo de domínio, que está inserido na disciplina de modelagem de negócio do RUP. Esse modelo descreve os conceitos e relacionamentos relativos a uma área de interesse. É nesse aspecto que ontologias e modelos de domínio assemelham-se. É então proposta a geração da ontologia do problema através do modelo de domínio. Com a ontologia integrada ao processo de desenvolvimento, tem-se uma maneira de conectar semanticamente diversos recursos, utilizando técnicas de inteligência artificial, segundo Kogut[KOG02].

Ao final do processo de desenvolvimento de software obtém-se um produto. Após este ter sido implantado, é dado início a uma nova fase do ciclo de vida, que é a de manutenção. Tudo aquilo que foi aprendido e produzido durante o processo de desenvolvimento, é extremamente necessário para que se possa mantê-lo. A seguir, serão abordadas as características de um projeto de manutenção e conceitos como rastreabilidade e análise de impacto que estão relacionados à maneira como os artefatos gerados durante o projeto podem ser acessados satisfatoriamente.

### **2.3 PROJETO DE MANUTENÇÃO DE SOFTWARE**

A manutenção de software pode ser definida como: “A alteração de um produto de software depois de entregue para corrigir falhas, para melhorar a performance ou outros atributos, ou adaptar o produto para um ambiente diferente”, segundo IEEE[IEE98]. Os projetos de manutenção são classificados em quatro categorias por Pressman[PRE95]: preventiva, que busca melhorar a confiabilidade ou manutenções futuras; adaptativa, que acompanha as evoluções tecnológicas; corretiva, que visa corrigir erros não encontrados na fase de teste, podendo esses ocorrerem na especificação, projeto ou implementação; e evolutivas que visam continuar satisfazendo as necessidades do usuário.

O padrão homologado pela IEEE para manutenção de software determina as fases que devem ser seguidas para realizar uma alteração no sistema. Após receber o requisito de mudança, esse deve ser analisado, ter seu projeto desenvolvido, realizar a implementação, fazer o teste de regressão do sistema e o teste de aceitação e por fim, a entrega ao cliente. A existência de um processo para manutenção de software não significa que essa será menos problemática. Diversos autores, tais como Bohner[BOH96], Ajila[AJI95] e Fay[FAY85], caracterizam a manutenção de software com geralmente sendo cara, difícil e demorada.

Uma das razões para esse tipo de afirmação é a dificuldade de entendimento do negócio e de sua posterior implementação por parte de uma pessoa que não participou do projeto de desenvolvimento. Mesmo profissionais que participaram do projeto tem dificuldade de recordar alguns detalhes deste ou de sua codificação. Essa situação é agravada pela falta de documentação ou pela sua não atualização ao longo do processo de desenvolvimento ou mesmo quando o software já está em manutenção. Uma outra razão está ligada à quantidade de artefatos do sistema que serão afetados por uma mudança.

Relacionado a esse último motivo, estão as definições de rastreabilidade e de análise de impacto de uma mudança dadas por Bohner[BOH96]. O primeiro pode ser definido como “a possibilidade de rastreamento entre os artefatos de software e seus componentes que são gerados e modificados durante o ciclo de vida do produto”. Já a análise de impacto pode ser definida como “a identificação das conseqüências de uma alteração ou estimativa do que é necessário para realizar uma mudança”. Esses aspectos serão apresentados a seguir, considerando as mudanças relativas a manutenções corretivas e evolutivas, já que essas geram requisitos de mudança funcionais, que são aqueles que representam as funcionalidades do sistema.

## 2.4 RASTREABILIDADE

Outras definições de rastreabilidade de requisitos foram dadas por Pearson[PEA96]: “rastreamento de requisitos é a habilidade de rastrear os requisitos do usuário através do processo de desenvolvimento, desde o levantamento de requisitos até a entrega do produto de software”. Já Gotel e Finkelstein[GOT94] defendem que “o rastreamento de requisitos é a habilidade de descrever e seguir a vida de um requisito, tanto para frente quanto para trás, ou seja, desde a sua origem, passando pela especificação e desenvolvimento, até sua entrega e uso; através de períodos de refinamento e iterações em qualquer uma dessas fases”.

Através dessas definições, pode-se concordar com Palo[PAL03], que diz que uma das importâncias de se ter um método para que um requisito seja rastreado é que quando este necessitar de alteração, encontre-se todos os artefatos ligados à ele e que serão afetados pela mudança. Outra característica importante no rastreamento, é que este pode dar toda a informação sobre as justificativas, decisões importantes e suposições sobre o requisito.

Foi estudado por Castor[CAS04] fatores que justificam a utilização do rastreamento de requisitos. Dentre os mais importantes estão:

- a) Qualidade de software: uma das definições é dada por Brooks[BRO87], que diz que a qualidade é a conformidade dos requisitos. Dessa forma, normas tais como *Capability Maturity Model for Software* (CMM) exigem disciplinas de rastreamento de requisitos. No CMM, esta consta no nível dois de maturidade;
- b) Análise de impacto: no decorrer do processo de desenvolvimento de software, muitas vezes ocorrem solicitações de mudanças de requisitos. Com o uso do rastreamento de requisitos é possível verificar quais artefatos serão impactados

pela mudança, podendo ser analisado se essa é viável ou não, uma vez que o esforço e tempo para ser realizada podem ser estimados;

- c) Manutenção de software: tendo em vista os projetos de manutenção de um software, o rastreamento dos artefatos torna-se essencial, uma vez que custos e esforços para efetivar a alteração podem ser determinados com maior precisão. Sem um método para rastrear os artefatos que serão afetados, são necessários a opinião e conhecimento de projetistas ou programadores, que muitas vezes podem não mais fazer parte da equipe de manutenção do sistema.

Segundo Gotel e Finkelstein[GOT94], apesar do número de ferramentas que suportam o rastreamento de requisitos terem aumentado, a teoria necessária para que sejam utilizadas não tem sido amplamente estudadas. Com isso, a maioria dessas ferramentas descobre e grava a maior quantidade de informação possível sobre o processo de engenharia de requisitos e faz relacionamentos entre essas informações para poder recuperá-las. O problema, ainda segundo Gotel e Finkelstein[GOT94], é que isso pode levar a uma dificuldade de usar essa informação, ainda mais se não existir um teste de usabilidade justificado destas, para que os usuários tenham acesso efetivo e satisfatório.

Alguns tipos de rastreabilidade foram estudados por Brooks[BRO87]. Um requisito pode ser rastreado pela sua direção e evolução, entre outros:

- a) Direcional: pode ocorrer “pra frente” (rastreamento para frente é a habilidade de rastrear um requisito para componentes de projeto ou implementação) ou “para trás” (rastreamento para trás é a habilidade de rastrear um requisito para sua fonte, isto é, para uma pessoa, instituição, lei, argumento). O primeiro tipo de rastreamento é utilizado quando há necessidade de saber o impacto da alteração. Já o segundo, é usado quando, dada uma alteração, necessita-se compreendê-la desde sua origem;
- b) Evolutiva: um rastreamento do tipo pré-especificação é utilizado quando, ainda na fase de construção da especificação, é necessário localizar as fontes dos requisitos ou pessoas responsáveis por este. Um tipo de rastreamento pós-especificação é realizado depois de todos os requisitos do sistema terem sido levantados e é usado, por exemplo, para relacionar todos os planos de testes desenvolvidos para atacar aquele requisito em específico e então determinar se estes também precisam ser modificados.

Ao longo dos anos, foram realizadas pesquisas com o intuito de propor modelos de rastreabilidade de requisitos. Segundo Brooks[BRO87], o rastreamento de requisitos tradi-

cional consiste em estabelecer relacionamentos bidirecionais entre requisitos e demais artefatos produzidos num processo de desenvolvimento de software. Podem-se citar como trabalhos relevantes os propostos por Gotel e Finkelstein[GOT94], Ramesh e Jarke[RAM01] e Toranzo e Castro[TOR99].

Definindo-se um modelo para rastrear os requisitos ao longo do processo de desenvolvimento de software, e também na fase de manutenção do produto, é possível determinar o impacto que uma mudança terá no sistema. Ou seja, é possível determinar previamente todos os artefatos que serão alterados, qual o esforço para esse trabalho e quanto tempo será necessário. Na seção seguinte será abordada a disciplina de análise de impacto, visando a diferenciação entre esta e a rastreabilidade.

## 2.5 ANÁLISE DE IMPACTO

A diferença entre rastreabilidade e análise de impacto de uma mudança é bastante tênue, mas significativa. Como foi descrita anteriormente, a rastreabilidade preocupa-se em definir a relação existente entre os artefatos do software. A análise de impacto, por sua vez, determina quais ações devem ser tomadas para realizar a alteração, podendo estimar quais serão as conseqüências desta. Apesar da diferença, ambas as abordagens estão intimamente relacionadas, uma vez que a metodologia utilizada para analisar o impacto é baseada na forma como a rastreabilidade foi construída.

Ajila[AJI95] citou as principais razões para realizar uma análise de impacto:

- a) Estimar o custo de uma mudança. Ou seja, se uma alteração irá afetar várias partes disjuntas do software, talvez seja melhor não realizá-la ou reexaminá-la;
- b) Entender a razão e o relacionamento entre o escopo da alteração e a estrutura do sistema. Ou seja, através do rastreamento dos artefatos que serão afetados, conhecer quais deles realmente serão modificados e quais têm chances de ser;
- c) Gravar o histórico de uma mudança e avaliar se esta foi realizada com qualidade, não comprometendo outras funcionalidades do sistema;
- d) Saber em quais partes do software deverá ser realizado teste de regressão, de forma a garantir a qualidade do sistema, principalmente nas seções críticas deste.

A análise de impacto pode ser vista através de dois ângulos, propostos por O'Neal[ONE03]: através da abordagem de análise de dependência e da análise de rastreabilidade. A primeira é a análise do relacionamento entre partes do código fonte, visando descobrir quais partes deste serão afetadas pela alteração. Essa abordagem permite que as relações sejam estabelecidas ao verificar o código fonte. A segunda preocupa-se em analisar o relacionamento entre artefatos produzidos nas diversas fases do processo de desenvolvimento do sistema, o que permite uma visão mais ampla do impacto da mudança no sistema como um todo. Para que seja possível o uso dessa abordagem é necessário que os artefatos tenham sido previamente relacionados.

Diversos autores propuseram modelos para determinar a análise de impacto de uma mudança, que são apresentados na subseção seguinte.

### **2.5.1 Modelos de Análise de Impacto**

Bohner[BOH96] relatou diversos modelos de manutenção de software, a maioria proposto na década de 80. Como mais relevantes podem-se citar os de Boehm, que consiste em três fases: entender o software, modificá-lo e revalidá-lo. Parikh enfatiza a identificação dos objetivos antes de entender o software, depois modificar o código e por último validar a alteração. O modelo de Osborne é focado na gerência das atividades de manutenção embora não proponha métricas para analisar o impacto de uma mudança. Patkow propôs um modelo focado na identificação e especificação dos requisitos de mudança. Após a mudança ser compreendida e localizada, são feitos: o projeto da alteração, implementação e posterior validação do sistema. A principal característica do modelo é a preocupação com a especificação e localização de onde a mudança deve ser feita. Lewis e Henry afirmaram a necessidade de coletar métricas ao longo do desenvolvimento do software para que o produto seja mais manutenível.

De forma genérica, todos os modelos atuais pesquisados trabalham com o fluxo mostrado na figura 2. Como entrada, tem-se todos os artefatos do ciclo de vida do software, ou seja, todos os artefatos produzidos durante o processo de desenvolvimento mais aqueles que foram incluídos ou alterados durante o projeto de manutenção. Esses objetos são analisados, de forma a determinar se será afetado ou não pela alteração requerida. Ao final do fluxo, tem-se a lista de todos os artefatos que deverão ser modificados.

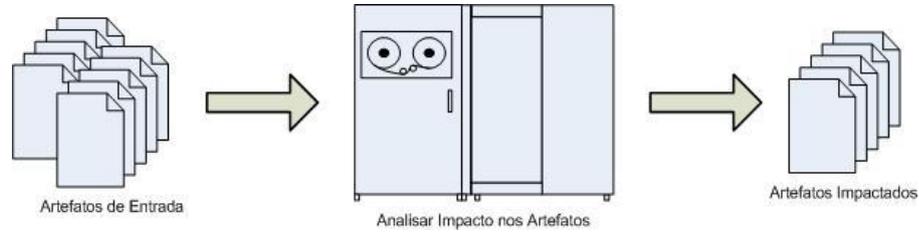


Figura 2 – Fluxo para determinar análise de impacto de mudança.

Este trabalho detém-se no processo de analisar quais objetos do ciclo de vida do software serão impactados por um requisito de mudança e como esses são encontrados, através da rastreabilidade. Serão abordados os métodos considerados mais relevantes para a presente proposta. São eles o de Bohner[BOH02] e o de O’Neal[ONE03] que serão expostos a seguir.

### 2.5.1.1 Modelos de Bohner

O método de análise de impacto de uma mudança proposto por Bohner[BOH02] tem caráter iterativo. A figura 3 apresenta o modelo.

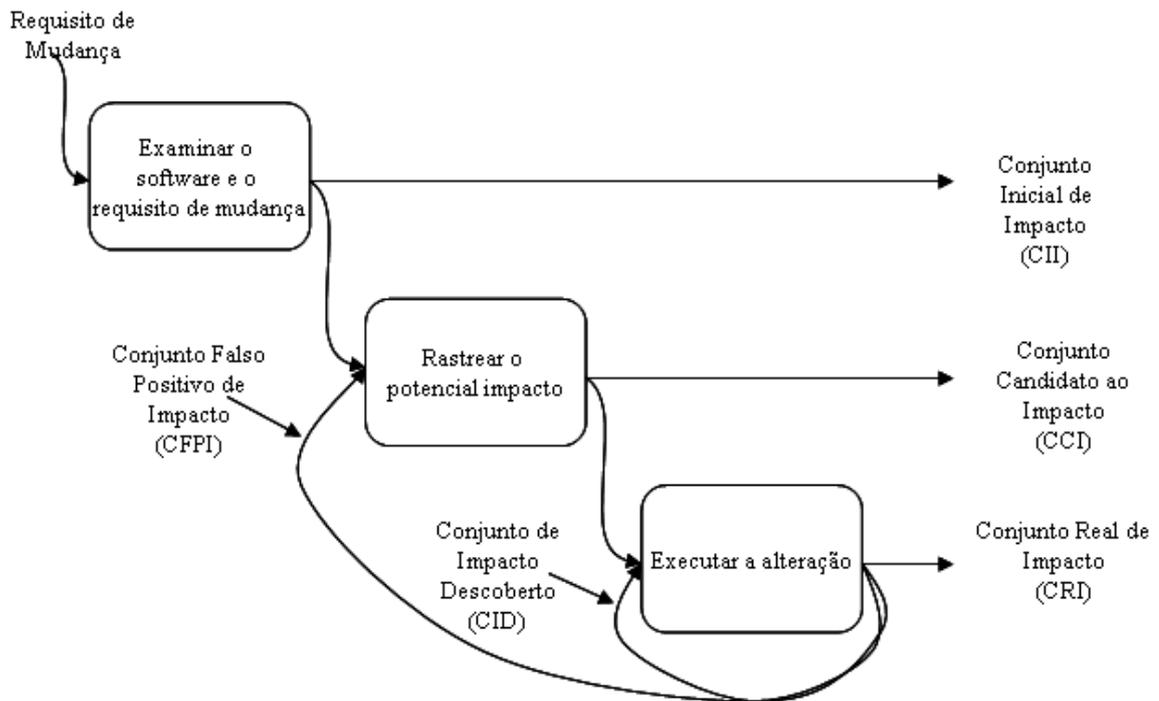


Figura 3 – Processo de análise de impacto proposto por Bohner.

O conjunto inicial de impacto (CII) são os primeiros objetos do ciclo de vida do software (OCV) que se supõem serão afetados pela mudança. Esses normalmente são levanta-

dos durante a análise do requisito de mudança. O conjunto candidato ao impacto (CCI) são os objetos que provavelmente serão afetados pela mudança. Estes são determinados quando a mudança é rastreada entre os artefatos. O conjunto real de impacto (CRI) são os OCV's que realmente serão alterados. Esse processo é considerado iterativo, pois enquanto uma alteração está sendo executada, podem-se encontrar outros objetos que são impactados que não foram previstos anteriormente. Esses novos objetos impactados formam o conjunto de impactos descobertos (CID), que representa uma sub-estimativa. Em oposição a este, tem-se o conjunto falso positivo de impacto (CFPI), representando uma super-estimativa.

De posse dessas variáveis, podem-se determinar quais objetos serão alterados, ou seja, o CRI é a soma do CCI, ou os objetos candidatos a alteração, e CID, isto é, os artefatos que foram encontrados posteriormente, menos CFPI, ou seja, os artefatos que inicialmente pensou-se que seriam afetados. A fórmula 1 mostra como é realizado esse cálculo.

$$CRI = CCI + CID - CFPI$$

Fórmula 1 – Artefatos que serão alterados

O objetivo do processo é encontrar o conjunto de objetos CCI, através de técnicas de rastreabilidade manuais ou automáticas, que seja o mais próximo possível do conjunto de artefatos realmente modificados, o CRI.

Bohner distingue o impacto em dois tipos: direto e indireto. Isso porque uma mudança pode ter o efeito de uma onda, afetando objetos que não tem uma ligação direta ou óbvia com a alteração solicitada. O impacto direto (ou impacto de primeiro nível) ocorre quando o objeto afetado é relacionado a outro diretamente. Pode ser obtido a partir de um grafo de conectividade de uma matriz de dependência.

O impacto indireto (ou impacto em  $n$ -níveis) ocorre quando o objeto afetado está relacionado a um conjunto de dependências. Esse conjunto é determinado pelos  $n$ -níveis de relações intermediárias entre os diversos artefatos e o objeto afetado. Esses objetos podem ser obtidos por um grafo de alcance de uma matriz de dependência.

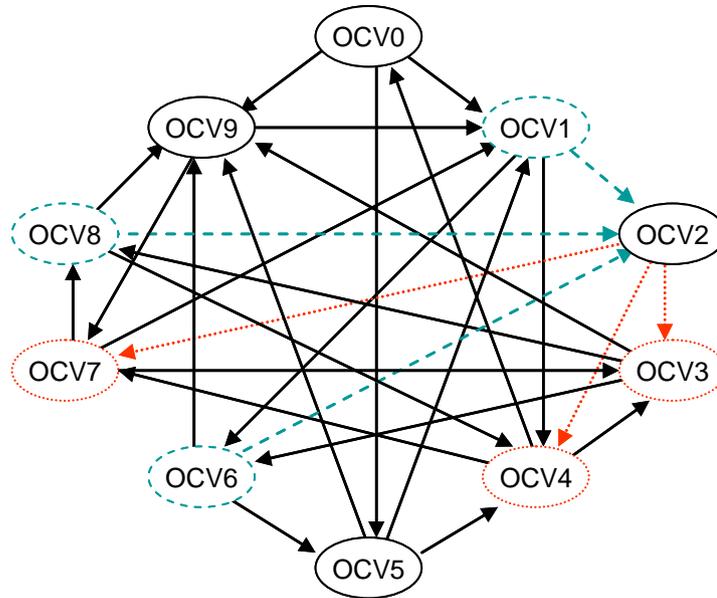


Figura 4 – Grafo com relacionamento entre objetos do ciclo de vida do software (OCV)

A figura 4 representa um grafo com relacionamento entre os objetos. A partir de um grafo como esse, pode-se extrair uma matriz de conectividade, representando quais objetos são diretamente afetados por um artefato. Por exemplo, o objeto OCV2 tem impacto direto dos objetos OCV1, OCV8 e OCV6, que estão representados pelas linhas tracejadas. Já os objetos OCV3, OCV4 e OCV7 são dependentes dele, representados pelas linhas pontilhadas. A tabela I apresenta a matriz de conectividade para o grafo da figura. 4.

Tabela I – Matriz de conectividade

	OCV0	OCV1	OCV2	OCV3	OCV4	OCV5	OCV6	OCV7	OCV8	OCV9
OCV0		X				X				X
OCV1			X		X		X			
OCV2				X	X			X		
OCV3							X		X	X
OCV4	X			X				X		
OCV5		X			X			X		
OCV6			X			X				X
OCV7		X		X					X	
OCV8			X		X					X
OCV9		X			X			X		

Segundo Bohner[BOH02], pode-se gerar um grafo de alcance a partir de uma matriz de conectividade, como a mostrada na tabela I, usando algoritmos de fechamento transitivo. O problema é que, embora um grafo de alcance possa indicar o potencial impacto de um artefato em outro, esse grafo tende a mostrar que todos os objetos estão conectados, direta ou indiretamente. Por isso, é necessário outro tipo de informação estrutural, como por exemplo, qual o nível de relação entre os objetos. Observando o grafo da figura 4 pode-se perceber que

o OCV1 e OCV8 têm nível três de relação, ou seja, é necessário passar por outros dois objetos (OCV4 e OCV7) e três conectores para se relacionarem. Com essa relação, o autor afirma que os artefatos que possuem relacionamento direto, potencialmente serão impactados e os artefatos mais distantes provavelmente não sofrerão alteração.

O autor faz distinção entre duas abordagens, uma baseada na estrutura e outra na semântica. O que foi descrito até aqui faz parte da abordagem estrutural. Segundo ele, a importância da abordagem semântica se dá com a intenção de aumentar a precisão ao determinar os objetos afetados. Tendo como base a orientação a objetos e os artefatos produzidos ao longo do processo de desenvolvimento, o autor sugere que sejam analisados os nomes de cada um desses artefatos e de suas relações. A técnica proposta por ele extrai os substantivos (dos dados dos objetos) e verbos (de funções e mensagens, por exemplo) para que se obtenha a lista de artefatos candidatos. Um dos pontos fracos desse método é que quando os nomes dados aos artefatos são ambíguos ou as convenções para esses são fracas, não é possível realizar o rastreamento através da semântica.

Na proposta de Bohner[BOH02], não são apresentados resultados que validem o método por ele apresentado. Sem a validação, não é possível certificar-se de que realmente todos os artefatos impactados por uma mudança são encontrados através desse método.

### 2.5.1.2 Metodologia baseada em rastreabilidade

A metodologia proposta por O’Neal[ONE03], está inserida no processo de desenvolvimento de software, mais especificamente na disciplina de Engenharia de Requisitos. Quando existe a solicitação de mudança de um requisito, esse é analisado de forma a estabelecer o impacto nos demais objetos do projeto a ele relacionados. A metodologia de análise de impacto para requisitos de mudança denominada *Trace-based Impact Analysis Methodology* (TIAM), avalia um conjunto de requisitos de mudança. A finalidade é prever o efeito que um conjunto de alterações terá em requisitos já estáveis, onde provavelmente o trabalho já esteja finalizado. Algumas premissas são assumidas:

- a) Requisitos funcionais do produto são rastreáveis;
- b) É possível fazer a rastreabilidade “para frente”, ou seja, a partir do requisito pode-se chegar a outros artefatos, até, por exemplo, ao código fonte;
- c) Todos os artefatos estão disponíveis e estão atualizados;

- d) A equipe do projeto ter informado qual é a influência de um artefato sobre outro, bem como a sua complexidade;
- e) O processo de desenvolvimento utilizado permita a construção por módulos, dando preferência para um paradigma de orientação a objetos.

O rastreamento entre os objetos é representado por um conjunto denominado pelo autor como *Work product Requirements trace Model* (WoRM), que é composto por outros dois conjuntos: *Work product Information Model* (WIM), que contém todos os artefatos produzidos ao longo do projeto e o *Requirements change Information Model* (RIM), composto por todos os requisitos de mudança. Os modelos WIM e RIM relacionam-se, de forma que toda alteração solicitada para um requisito de mudança aponta para o referido requisito. Por sua vez, o requisito aponta para outro artefato e assim sucessivamente. A figura 5 demonstra esses relacionamentos:

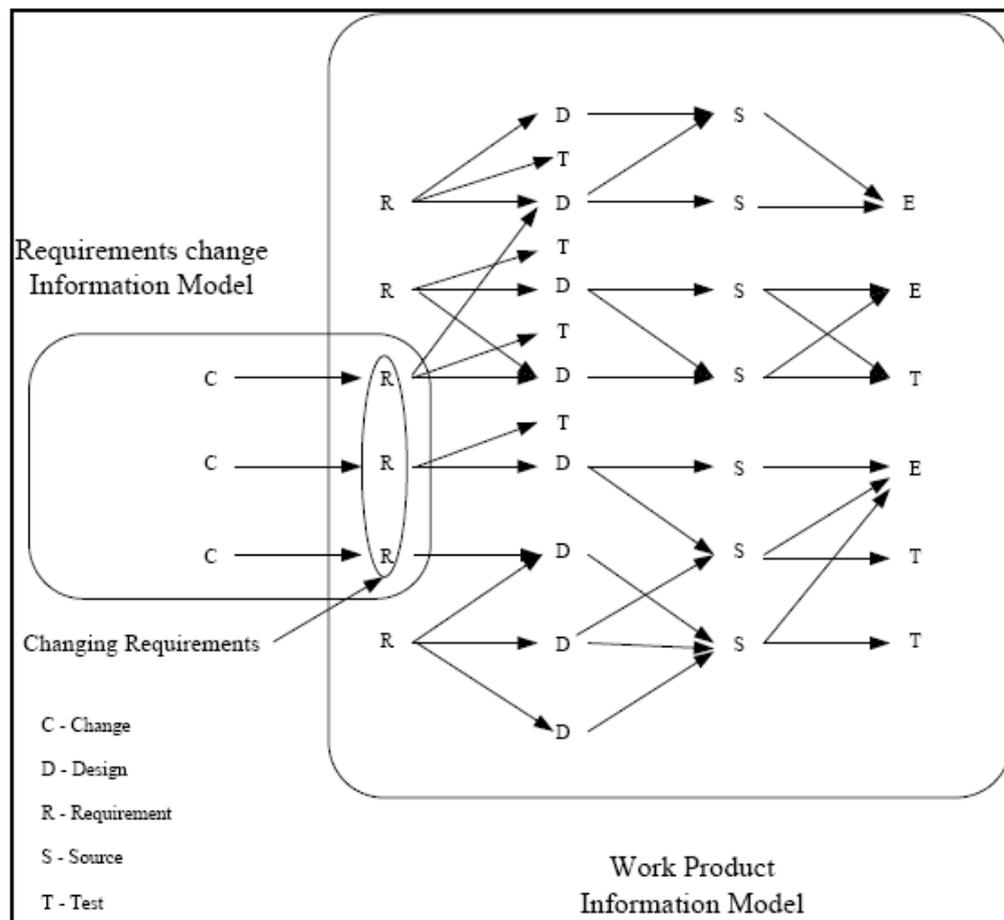


Figura 5 – Modelo de rastreamento entre os artefatos do sistema

Os produtos de trabalho, referidos pelo autor, possuem alguns atributos externos, que irão auxiliar na determinação do impacto. Os atributos usados para calcular o peso de cada artefato são: a complexidade, que mede o grau de dificuldade para desenvolver o objeto;

qual o esforço para produzi-lo, medido em homens/hora e em qual fase de desenvolvimento encontra-se o artefato, uma vez que é mais difícil alterá-lo nas fases finais de desenvolvimento, segundo Schach[SCH99]. Deve-se ainda, estabelecer o grau de influência que um artefato exerce sobre outro. Além da influência entre os artefatos, é preciso determinar a influência que a mudança tem sobre o requisito a ela associado. A figura 6 mostra como os produtos de trabalho se relacionam e qual o grau de influência entre eles.

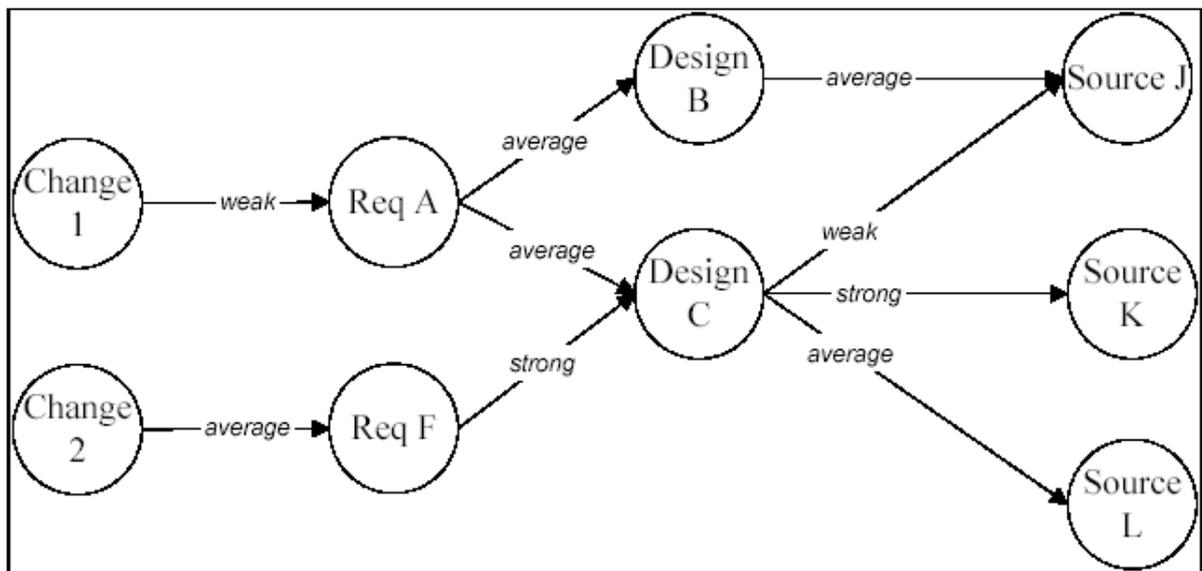


Figura 6 – Relacionamento entre mudanças, artefatos e grau de influência entre eles.

O conjunto de graus de influências de mudanças e de associações entre artefatos pode ser configurado conforme o projeto e dados históricos da empresa. No exemplo da figura 6, tem-se que o conjunto de graus de influência é:  $\{weak; average; strong\}$ . A esses graus, são associados valores, também configuráveis. De posse desses, é calculada a influência de cada arco. Esta deve ser normalizada quando dois artefatos se ligam a um outro, na figura 6 “Design B” e “Design C” ligam-se a “Source J”. A figura 7 exemplifica as influências normalizadas entre os objetos e o peso de cada artefato do sistema.

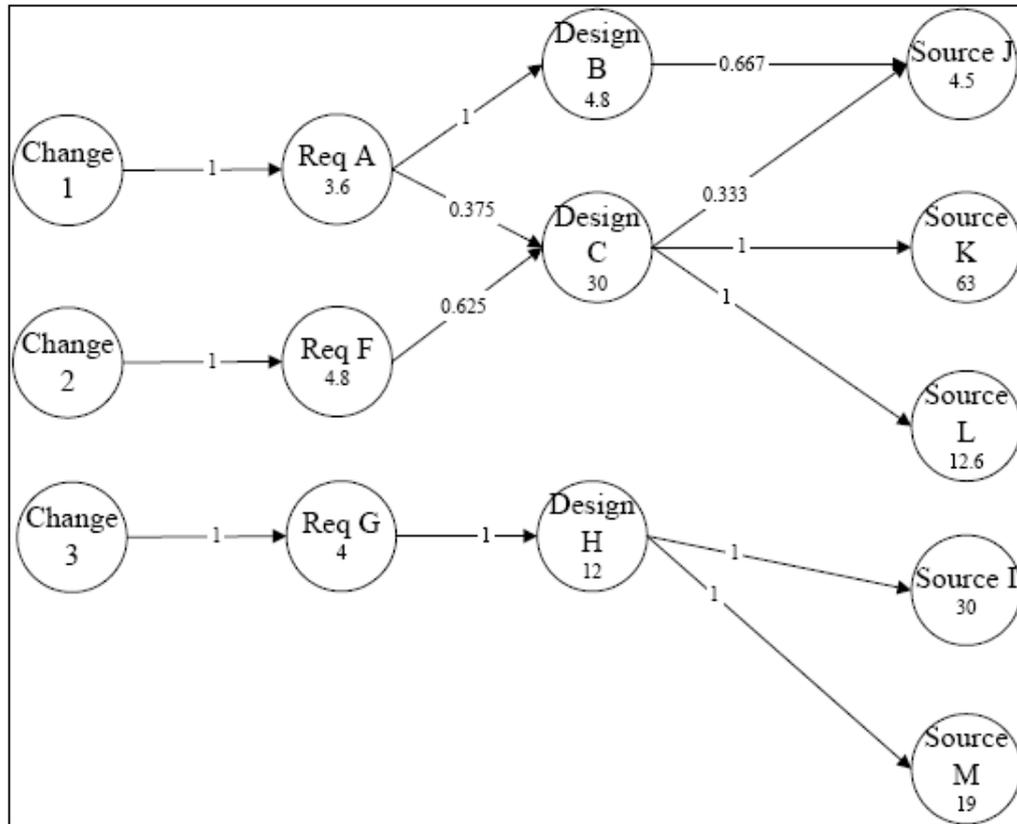


Figura 7 – Relacionamento entre artefatos e requisitos de mudança.

De posse dos modelos WIM e RIM, dos pesos dos artefatos e dos valores de influência entre eles, calculam-se as métricas de impacto para cada um dos requisitos de mudança, representando a probabilidade de impacto. As métricas são então agrupadas em conjuntos conforme a similaridade de impacto, a partir de um grau de similaridade. Por exemplo, se o ponto de corte de similaridade for de 70%, todos os requisitos de mudança que obtiverem a métrica maior ou igual a esse ponto, formarão um conjunto. As demais serão armazenadas em outro. Em de cada um desses conjuntos, as métricas são ordenadas de forma decrescente. A aplicação da metodologia TIAM caracteriza as mudanças baseadas no nível de impacto de cada uma delas.

A metodologia de O'Neal[ONE03] é interessante, pois possui mecanismos para determinar os artefatos que são potencialmente afetados por uma mudança. Além disso, seu uso foi experimentado e validado com alunos de graduação de um curso de engenharia de software da Universidade de Lousiana, Estados Unidos. O ponto fraco está na rastreabilidade entre os artefatos, uma vez que é estruturalmente realizada e não possui semântica. O autor também não propõe um método para calcular a influência que um artefato exerce sobre outro, sendo esta atribuída a um especialista, sem parâmetros nos quais possa se basear.

Na próxima seção, será abordada a Engenharia de Software Experimental. Tem-se como objetivo apresentar essa linha de pesquisa, que está inserida na engenharia de software, visando sua utilização para, através de um experimento, analisar qual é o impacto de uma mudança rastreada através de conceitos de uma ontologia.

## 2.6 ENGENHARIA DE SOFTWARE EXPERIMENTAL

A engenharia de software experimental está inserida na discussão sobre classificar a engenharia de software como ciência ou engenharia. Segundo Travassos, Gurov e Amaral[TRA02], essa questão está ligada ao significado que se dá a software, estabelecendo-o como produto ao considerar seu processo de criação, que possui características de produção ou engenharia, ou que pode considerar a necessidade de melhoria contínua da qualidade do processo e do produto, sendo então caracterizado como ciência.

Em Basili[BAS96], o autor faz uma comparação da engenharia de software com outras áreas, tais como física, medicina e manufaturas. Segundo ele, as duas primeiras, diferentemente do que acontece com o produto final da engenharia de software, não podem ter a essência de seus produtos alterados, já na manufatura sim. E, diferente da manufatura, o software é desenvolvido e não produzido, ou seja, o mesmo software não será produzido mais de uma vez da mesma maneira.

Ainda segundo o autor, existem características intrínsecas ao software que o diferem de qualquer outra ciência. Uma delas é que mesmo aumentando o nível de abstração, o desenvolvimento de software depende de pessoas, com seus aspectos individuais e criatividade únicas. Isso implica que um mesmo experimento realizado com grupos distintos de pessoas poderá ter resultados diferentes.

Conforme proposto por Wohlin, Runeson, Host, Ohlsson, Regnell e Wesslen[WOH00], existem diferentes métodos para experimentos na área de engenharia de software: científico, de engenharia, experimental e analítico. O científico pode ser usado para tentar entender um processo com o intuito, por exemplo, de construí-lo, extraíndo do mundo algum modelo que possa explicar o fenômeno e avaliá-lo. O método de engenharia tem uma abordagem evolutiva, analisando algum modelo existente e modificando-o para tentar melhorá-lo. O método analítico não precisa de um projeto experimental no sentido estatístico, pois possui uma base analítica para validar a teoria formal sugerida.

O método experimental (de maior importância para o presente trabalho), segundo Travassos, Gurov e Amaral[TRA02] inicia-se com o levantamento de um modelo novo, existente ou não, e estuda o efeito do processo ou produto a partir desse modelo, podendo ser analisado quantitativa ou qualitativamente.

Esse tipo de método tem como base as variáveis, os objetos, os participantes, o contexto do experimento, as hipóteses e o tipo de projeto do experimento. As variáveis podem ser dependentes, que se referem à saída do processo, ou independentes, que se refere à entrada do processo.

O objeto é usado para verificar a causa e efeito numa teoria e juntamente com o sistema de medição, compõem a instrumentação do experimento. Os participantes são as pessoas selecionadas para realizar o experimento. É necessário observar que, para um experimento ser considerado generalizado para uma população alvo, os participantes escolhidos devem ser representativos. O contexto do experimento refere-se as condições onde este será aplicado, ou seja, se será *in-vitro* (laboratório) ou *in-vivo* (projeto real); alunos ou profissionais; problema de sala de aula (*toy example*) ou problema real; específico (contexto particular) ou geral (para toda Engenharia de Software).

Talvez o aspecto mais importante do experimento dê-se pela escolha das hipóteses, que são divididas entre a hipótese nula (que é a principal) e as hipóteses alternativas. A hipótese nula nega o relacionamento estatisticamente significativo entre a causa e o efeito. Sendo assim, o objetivo do experimento é negá-la a favor de uma hipótese alternativa.

Outros aspectos centrais de um experimento correspondem à medição e a validade. Através delas, o mapeamento entre o mundo experimental e o mundo formal ou relacional é feito. A validade diz respeito aos resultados do experimento e pode ser classificada em validade de conclusão, onde se valida o relacionamento entre o tratamento (teste estatístico, escolha dos participantes, confiabilidade da medida) e o resultado do experimento; validade interna, se o resultado é causal e não foi manipulado por influência de variáveis não controladas; validade de construção, onde são avaliados os aspectos importantes do experimento e o fator humano, no que diz respeito ao comportamento dos participantes e do experimentador; e a validade externa, onde é limitada a habilidade de generalizar os resultados de um experimento para prática industrial.

A seguir, serão apresentadas as fases necessárias para realização um experimento, propostas por Travassos, Gurov e Amaral[TRA02]:

- a) Definição: o experimento é expresso em termos de problemas e objetivos. Os objetivos são separados em objetivo global, objetivo da medição e objetivo do

estudo. É sugerido o uso da técnica GQM (*Goal, Question, Metric*) para determinar os objetivos;

- b) Planejamento: o projeto do experimento é determinado, a instrumentação é considerada e os aspectos da validade do experimento são avaliados. É preciso realizar a seleção do contexto, formular as hipóteses, selecionar as variáveis, selecionar os participantes, projetar o experimento, fazer a preparação conceitual da instrumentação e fazer a consideração da validade do experimento;
- c) Execução: os dados são coletados. Os participantes devem passar por uma preparação antes do início da execução. Também é feita a validação preliminar dos dados;
- d) Análise e interpretação: são realizadas a análise e interpretação dos dados e feita à conclusão sobre a possibilidade de rejeição da hipótese nula. Os passos que devem ser executados são: validação dos dados, estatística descritiva, aplicação do teste estatístico, análise quantitativa e qualitativa e verificação das hipóteses;
- e) Apresentação e empacotamento: os resultados obtidos são apresentados e empacotados. O empacotamento é importante, pois é necessária a repetição do experimento, para poder avaliar os resultados comparando-se com o experimento original. Atualmente, essa fase do experimento não possui normas internacionais aprovadas.

### 3 MODELANDO A ANÁLISE DE IMPACTO DE MUDANÇA

Para viabilizar a análise de impacto de um requisito de mudança é necessário, primeiramente, conhecer o que foi produzido durante o processo de desenvolvimento do software e, principalmente, identificar o tipo de rastreabilidade utilizada. O conhecimento dos artefatos é necessário tendo em vista que cada um deles terá uma influência diferente no impacto que será estabelecido. Uma das premissas básicas para realizar a análise de impacto, é ter a possibilidade de rastrear os artefatos, uma vez que esta irá determinar, a partir do requisito de mudança, as implicações da alteração.

A proposta de uma metodologia de análise de impacto de mudança, utilizando a rastreabilidade semântica proporcionada pelo uso de ontologias, foi construída com base no estudo de outras propostas na área de análise de impacto. A metodologia de análise de impacto baseada em rastreabilidade, desenvolvida por O'Neal[ONE03], foi considerada a mais interessante, pois utiliza para relacionar os artefatos a estrutura de um grafo, que também é utilizado para representar uma ontologia.

O modelo de análise de impacto proposto nesse trabalho tem como base um processo de desenvolvimento que utilize a Engenharia Ontológica e construa a rastreabilidade semântica através da ontologia do sistema. Ainda como premissa para determinar o impacto de um requisito de mudança deve-se estabelecer a influência de um tipo de mudança na ontologia, a influência dos relacionamentos entre os conceitos da ontologia, a influência que um conceito da ontologia exerce sobre um artefato relacionado a ele e a influência do tipo do artefato, no que se refere a diagramas da UML. Estas influências serão detalhadas nesse capítulo.

O experimento, descrito no Apêndice A, foi aplicado para conhecer de que forma alterações nos artefatos do sistema se refletem na ontologia. Assim, foi possível perceber dois tipos de influência na ontologia: a influência por tipo de mudança, que está relacionada a alterações diretamente nos conceitos, e a influência do relacionamento entre os conceitos, que retrata a propagação de uma mudança. Essas duas influências serão amplamente abordadas a

seguir, demonstrando o cálculo do impacto para cada uma e analisando as variáveis que se relacionam a elas. Além dessas, serão também discutidas algumas variáveis utilizadas por O’Neal em sua metodologia, estabelecendo sua ligação com a presente proposta.

### 3.1 ENGENHARIA ONTOLÓGICA

A proposta de Noll[NOL05b] estende o RUP para que este passe a integrar em suas disciplinas a Engenharia Ontológica. Esta nova disciplina é responsável por construir a ontologia e atualizá-la ao longo do processo de desenvolvimento, visando manter a integridade das informações produzidas durante o projeto.

Baseado em propostas da área de Engenharia Ontológica, o autor propõe atividades que serão realizadas em cada fase da disciplina, a saber Projeto, Manutenção e Validação da ontologia. Essas atividades são executadas por um ator que também é responsável pelos artefatos produzidos e por organizar as iterações. A tabela II mostra como as atividades são distribuídas nas fases.

Tabela II – Atividades por fase da Engenharia Ontológica.

<b>Fase</b>	<b>Atividade</b>
Projeto	Definição do escopo da ontologia.
	Aquisição e explicação dos conceitos do domínio
	Formalização dos conceitos
Manutenção	Integração às ontologias existentes (se preciso)
	Definição de axiomas
Validação	Validação

Na fase de Projeto, a ontologia preliminar é gerada. Utiliza-se para isso o Modelo de Domínio, produzido durante a disciplina de Modelagem de Negócio do RUP. Neste modelo, os conceitos relevantes a uma área de interesse e seus relacionamentos são descritos.

A fase de Manutenção é caracterizada pelo refinamento da ontologia, onde se definem estruturas lógicas para esclarecer o modelo e podem-se integrar ontologias previamente existentes, caso seja identificada essa necessidade. Os conceitos da ontologia são refinados a medida que artefatos como Especificação de Requisitos, Modelos de Análise e Modelos de Projetos são desenvolvidos.

A fase de Validação deve ser realizada a cada iteração do ciclo de desenvolvimento com a finalidade de avaliar a integridade, a completude e a corretude da ontologia em relação ao modelo lógico do projeto.

A integração da Engenharia Ontológica com o RUP é dada pela figura 8 onde Noll[NOL05b] insere a nova disciplina entre as existentes originalmente. Segundo ele, o início dessa disciplina ocorre logo após o começo da disciplina de Modelagem de Negócio, pois é nela que se estabelece, em nível conceitual, o universo de discurso. Além disso, a maior parte das atividades dessa nova disciplina está concentrada entre as fases de iniciação e de elaboração uma vez que é nesse momento que os conceitos do domínio são definidos.

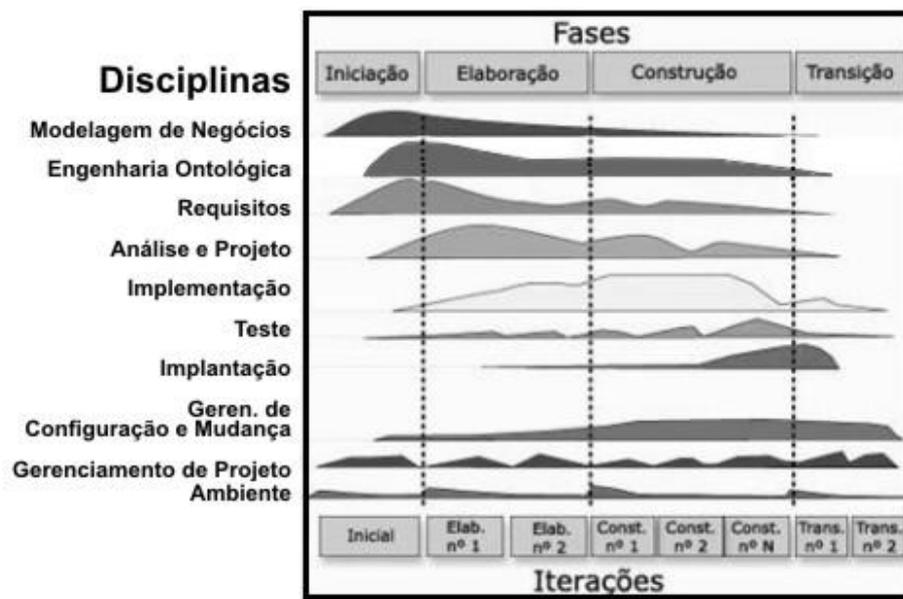


Figura 8 – Integração da disciplina de Engenharia Ontológica ao RUP

Para realizar a análise de impacto baseada na rastreabilidade por ontologias é imprescindível que a Engenharia Ontológica faça parte do processo de desenvolvimento. Com a ontologia sendo gerada, mantida e validada ao longo do ciclo de vida do projeto, é possível estabelecer elos de rastreabilidade entre os conceitos da ontologia e artefatos produzidos no decorrer do uso do RUP. A seção a seguir descreve como é estruturada a rastreabilidade com o uso de ontologia.

### 3.2 RASTREABILIDADE SEMÂNTICA ATRAVÉS DE ONTOLOGIA

Para que seja possível realizar a rastreabilidade ontológica, a proposta de Noll[NOL05b] utiliza a ferramenta ONTrace. Esta permite, através de mecanismos próprios, o rastreamento dos artefatos a partir da ontologia gerada para representar o domínio do problema.

Existem dois conceitos associados aos recursos do ONTrace: *model*, onde estão definidos todos os tipos previstos na UML; e *artifact*, que é um dos elementos produzidos no processo de desenvolvimento de software. Esse último possui duas propriedades associadas: *ontraceRecover*, onde são relacionados todos os conceitos da ontologia aos quais o artefato está relacionado e *ontraceArtifactsTypedBy*, que representa o tipo do artefato, definido pelo conceito *model*. Na figura 9 está a representação da rastreabilidade semântica, considerando os diversos tipos de artefatos produzidos ao longo do processo de desenvolvimento do software.

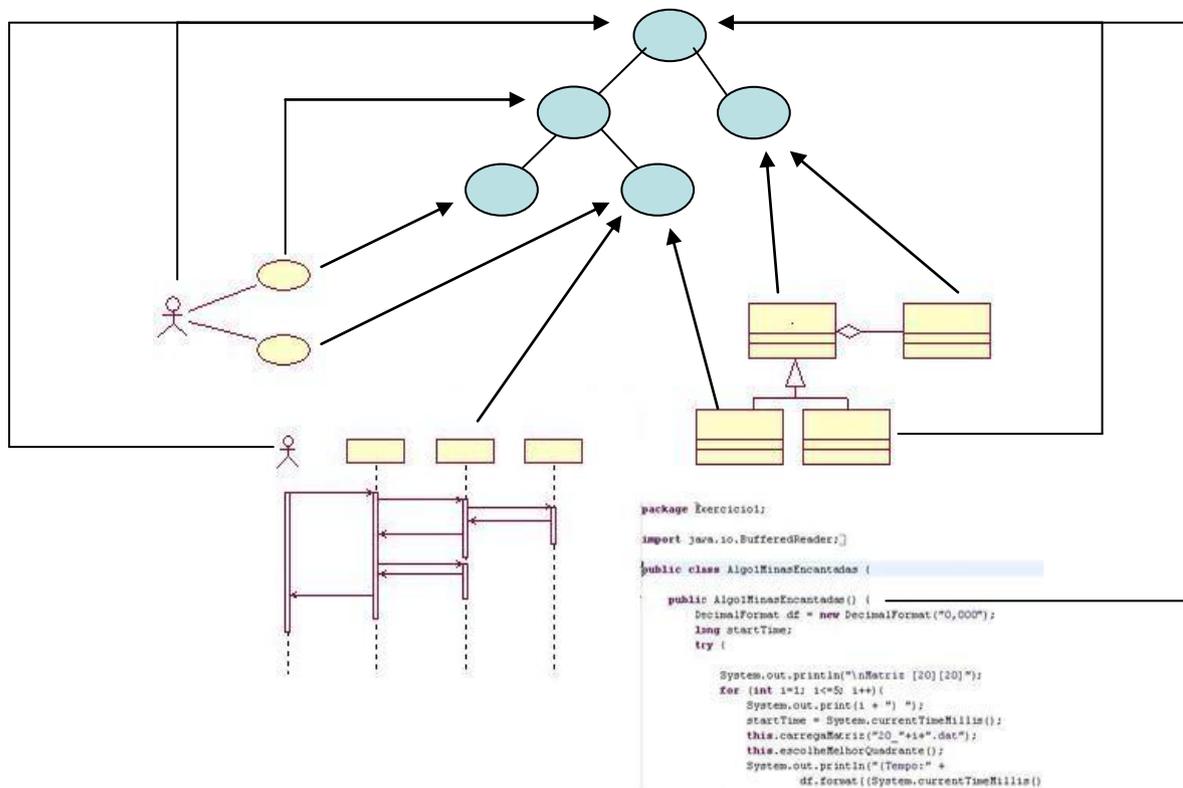


Figura 9 – Representação da rastreabilidade semântica através de ontologia

Através da ferramenta desenvolvida por Noll[NOL05b] é possível associar os conceitos da ontologia gerada aos elementos do modelo. Concluído esse processo de associa-

ção entre conceitos e artefatos, pode-se buscar o conjunto de elementos que estão relacionados a um mesmo conceito, possibilitando o conhecimento de todos os artefatos que serão impactados por uma alteração.

Utilizando o exemplo dado pelo autor, o caso de uso “Registrar compra”, pode estar associado a conceitos como “cliente”, “compra”, “produto”. Outro caso de uso, “Cadastrar cliente”, relaciona-se com o conceito “cliente”. Tem-se então um relacionamento entre os casos de uso indexados pelo conceito “cliente”.

O rastreamento de artefatos pode ocorrer em outros níveis, além do relacionamento direto apresentado no exemplo anterior. Se conceitos da ontologia possuírem, por exemplo, relacionamentos não taxonômicos, os artefatos associados a esses conceitos podem se relacionar. Supondo que o caso de uso “Cadastrar funcionário” esteja associado ao conceito “funcionário”, e o caso de uso “Cadastrar cliente” esteja associado ao conceito “cliente”, se os conceitos “funcionário” e “cliente” estiverem relacionados por uma propriedade, talvez um caso de uso seja afetado por uma mudança no outro.

Nas seções seguintes, serão estabelecidas as influências que elementos da ontologia possuem, dependendo do tipo de mudança, relacionamento e artefatos associados.

### **3.3 INFLUÊNCIA DA COMPLEXIDADE DE ARTEFATO**

Na metodologia TIAM de análise de impacto, a complexidade é uma das variáveis utilizadas para determinar o peso de um artefato. Um dos objetivos de mensurá-la é associar ao cálculo da probabilidade de impacto a dificuldade de manter o artefato, em termos, por exemplo, de estimativas de tempo, custo e esforço para realizar o requisito de mudança.

As seções seguintes expõem como a complexidade de artefatos pode ser estabelecida e são citadas outras formas de determiná-la, além dos modelos de custos propostos por Boehm[BOE01], cujo uso é sugerido por O’Neal. A intenção é demonstrar como essa variável é empregada em cada uma das propostas, na metodologia TIAM e na análise de impacto baseada em ontologias, e distinguir o seu uso em cada uma delas.

### 3.3.1 Complexidade de Artefato

A complexidade de artefatos é definida ou estimada pelo uso de métricas. Existem diversas propostas de métricas na bibliografia e o uso de cada uma delas está ligado a qual foi o processo de desenvolvimento usado no projeto, o paradigma de programação e a arquitetura do software utilizada, dentre outros fatores. Além disso, há a diferenciação entre métricas de processo e de produto.

No contexto dessa pesquisa é interessante verificar quais são as métricas utilizadas para produto de software, estabelecendo prioritariamente métricas para o paradigma de orientação a objetos relacionados a diagramas da UML. Ao final dessa subseção, será analisado o uso da complexidade de artefatos na proposta de análise de impacto baseada em ontologias.

#### 3.3.1.1 Métricas de produto de software

Nesses mais de trinta anos de pesquisa na área de métricas, a importância do desenvolvimento de projetos de software teve um crescimento bastante acentuado. O aumento dessa importância faz com que haja mais requisitos intrínsecos ao projeto. Desta forma, é necessário que exista um controle mais preciso tanto no processo de desenvolvimento quanto no produto a ser entregue. Isso se faz necessário, pois a maioria dos projetos atrasa, estouram o orçamento ou não possuem a qualidade desejada pelo cliente. Uma forma de evitar ou reduzir esse tipo de problema é conseguir estimar corretamente as variáveis envolvidas, tais como custo, esforço e prazo necessários para o desenvolvimento do projeto.

O processo de medição de software traz benefícios como modelar e entender o processo de desenvolvimento e o produto gerado, ajudar na gerência do projeto de software e melhorá-lo. Dentro desta realidade, a área de métricas de software, que é a medida quantitativa de atributos específicos do desenvolvimento de software, tem sugerido meios capazes de atingir os objetivos desejados.

Scotto [SCO04] classificou os tipos de métricas em métricas de processo, de produto e de recursos. O foco das métricas de produto está nas características do software em relação à qualidade de entrega. Segundo Sommerville [SOM03], medidas como tamanho e complexidade ciclomática não são facilmente relacionadas com atributos de qualidade, tais

como a facilidade de compreensão e manutenção, isto porque estão intimamente relacionadas com o processo de desenvolvimento e com a tecnologia aplicada. Uma forma de relacionar tais medidas e atributos à realidade do projeto que está sendo desenvolvido é coletar essas métricas e comparar os dados históricos dentro, por exemplo, de uma fábrica de software, procurando uma relação entre elas.

Sommerville [SOM03] propôs a classificação das métricas de produto em dinâmicas e estáticas. A primeira é relacionada diretamente à qualidade e são coletadas por medições realizadas por um programa em execução. Esse tipo de métrica pretende avaliar a eficiência e a confiabilidade de um programa. Já as métricas estáticas são medidas por representações do sistema, ou seja, pelo projeto, programa ou documentação, tendo uma relação indireta com atributos de qualidade. Essas métricas ajudam a medir atributos como a complexidade, a facilidade de compreensão e manutenção do software.

Outra proposta de classificação de métricas de produto foi dada por Mills [MIL99], que as separou em métricas de produto por tamanho, complexidade e qualidade. A medida por tamanho está diretamente relacionada ao produto e processo por meio do qual este foi desenvolvido; a medida por complexidade tem relação com o gerenciamento do processo de desenvolvimento; e a medida por qualidade que mensura características como a corretividade, confiabilidade e manutenibilidade, sendo essas as mais estudadas dentro de um tópico onde não existe consenso em uma única métrica para todos os itens de qualidade.

Além das classificações apresentadas, pode-se ainda classificar as métricas de produto em métricas orientadas a objetos. Pesquisas nesse sentido iniciaram na década de 90, sendo que algumas foram derivadas das métricas mais antigas, baseadas em programação estruturada, embora outras sejam exclusivas do paradigma de orientação a objetos. Dentro dessas pesquisas, os métodos propostos por Lorenz em 1993 e por Chidamber e Kemerer em 1994, são bastante referenciados. A primeira proposta é composta de um conjunto de onze métricas orientadas a objetos (OO). A segunda, também conhecida como métricas CK, possui um conjunto de seis métricas, voltadas para o projeto e complexidade. Essa proposta, dentre outras, serviu como base para as atuais métricas orientadas a objetos que utilizam diagramas da UML, segundo Kim[KIM02]. As métricas para esses diagramas serão apresentadas na seção seguinte.

### 3.3.1.2 Métricas para UML

Desde a sua criação, a UML vem sendo largamente utilizada no processo de desenvolvimento de softwares orientados a objetos. Segundo Kim[KIM02], uma vez que o número de artefatos produzidos utilizando a UML aumentou, existe a necessidade de mensurar as características desses objetos. Com isso, é possível obter uma noção exata, por exemplo, da complexidade do projeto já nas primeiras fases deste, que é o momento onde se começa a criação dos modelos UML. Uma das vantagens apontadas por McQuillan[MQU06], é de que, com o uso dessas métricas, a qualidade do software pode ser estimada ainda nas fases iniciais do ciclo de vida, onde o custo de se fazer uma mudança ainda não é alto.

Durante a pesquisa de trabalhos nesta área, não foi encontrado um modelo único de métricas para artefatos da UML. Cada proposta utiliza as métricas dentro do seu próprio modelo, com o objetivo de propor resultados mais precisos. Outra característica, segundo Genero[GEN03], é que a maioria das propostas trabalha com os diagramas estruturais, principalmente com o diagrama de classes. Tendo em vista essa realidade, será apresentada a seguir uma métrica de complexidade para esse diagrama.

#### 3.3.1.2.1 *Complexidade do Diagrama de Classes*

Dentro do estudo de métricas orientadas a objetos, a complexidade dos diagramas estruturais são os que possuem mais pesquisa, segundo Genero[GEN03]. Trabalhos propostos por Chidamberer e Kemerer [CHI94], as métricas CK, por Lorenz [LOR94], por Marchesi [MAR98] e por Fenton [FET00], dentre outros, estão relacionados a esse diagrama.

A proposta de Carbone [CAR02], denominada Fast&&Serious, propõe métricas para determinar a complexidade do diagrama de classes, ou somente de uma classe, já que considera esse artefato o mais importante da modelagem orientada a objetos. O método é composto por seis passos, sendo alguns opcionais, como mostra a tabela III.

Tabela III – Passos do método Fast&amp;&amp;Serious

Passo	Descrição
1	Determinar o tipo do método, se <i>i</i> ou <i>Serious</i> .
2	Calcular a complexidade de cada classe (CP) pertencente ao diagrama de classes que está sendo analisado.
3*	Avaliar o CPs para o diagrama de casos de uso.
4*	Avaliar o CPs para os diagramas de seqüência e de colaboração.
5*	Avaliar o CPs para o diagrama de estados.
6	Somar os CPs encontrados nos passos 2, 3, 4 e 5, obtendo o número de SLOC de todo o sistema.

Os passos que estão sinalizados na tabela III com (\*) não são obrigatórios. O primeiro passo determina qual método será utilizado, através da média da métrica do percentual de métodos com assinatura (PMS). Se o PMS for maior do que 70% é usado o método *Fast*, senão é usado o método *Serious*. A seguir, será apresentado apenas o método *Fast*, descrevendo o segundo passo da tabela III, pois a finalidade é conhecer como se determina a complexidade das classes de um diagrama de classes.

Os atributos de cada classe são classificados em três categorias, conforme a tabela IV. Após, é calculado o número de Pontos por Estado (SP), que é dado pela fórmula 2, onde *ListAtrib* é classificado pelo par  $\{atributo; tipo\}$  e *numAtrib* é o número de atributos da classe.

Tabela IV – Passos dos atributos no método Fast&amp;&amp;Serious

Complexidade	Definição	Peso
Leve	Atributos simples, como inteiro, string, etc.	1
Pesado	Atributos complexos desenvolvidos e testados, como atributos em pacotes pré-definidos.	3
Importado	Atributos complexos de uma classe não desenvolvida e não testada.	5

$$SP_{(c)} = \sum_{i=1}^{numAtrib_{(c)}} Peso(ListAtrib)$$

Fórmula 2 – Cálculo do número de Pontos por Estado

Após o cálculo de SP, os métodos são classificados em Triviais, que são métodos que possuem ao menos 80% de atributos do tipo leve e não tem atributos do tipo importado na sua assinatura, ou em Substanciais, que são os métodos que não foram classificados como triviais. No método *Fast*, que é o único que está sendo abordado nesse trabalho, todos os métodos são classificados como substanciais com um atributo do tipo importado. Caso o método *Serious* fosse aplicado, cada método da classe deveria ser classificado conforme os critérios apresentados. A complexidade do método é dada pela fórmula 3, onde *numLA*, *numHA* e *nu-*

$mIA$  são os números de atributos leves, pesados ou importados na assinatura do método, respectivamente. A variável  $T$  significa que o método é trivial ( $T=1; S=0$ ) e  $S$ , que é substancial ( $T=0; S=1$ ).

$$CM_{(m)} = \begin{matrix} T \\ S \end{matrix} \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 5 \end{pmatrix} \begin{pmatrix} numLA \\ numHA \\ numIA \end{pmatrix}$$

Fórmula 3 – Cálculo da Complexidade do Método

Em seguida, o Ponto por Comportamento (BP) é calculado para a classe, conforme a fórmula 4, onde  $numAss$  é o número de associações da classe e  $numMet$  é o número de métodos da classe.

$$BP_{(c)} = \left[ + numAss_{(c)} \right] \sum_{i=1}^{numMet_{(c)}} CM_{(m_i)}$$

Fórmula 4 – Cálculo dos Pontos por Comportamento

Por fim, é calculado, para cada classe, o Ponto por Classe (CP), que é a estimativa da complexidade desta, apresentada na fórmula 5.

$$CP_{(c)} = 2 \times SP_{(c)} + 3 \times BP_{(c)}$$

Fórmula 5 – Cálculo dos Pontos por Classe

Com esta estimativa é possível prever quando uma classe é mais complexa do que outra, caracterizando, possivelmente, uma maior dificuldade de manutenção das classes mais complexas. Um exemplo de aplicação desse método será dado na seção seguinte.

### 3.3.2 Uso da Complexidade de Artefato

Para utilizar a metodologia TIAM, o peso do artefato é uma das informações utilizadas para conhecer a métrica de impacto do requisito de mudança. Essa métrica possibilita que sejam determinadas outras estimativas com a finalidade de prever o que é necessário para efetuar cada alteração solicitada.

O peso do artefato é então dado pela fórmula 6, que além da complexidade do artefato ( $c_{(n)}$ ), relaciona o esforço ( $e_{(n)}$ ), que mensura o tempo necessário para produzir o artefa-

to, e a fase ( $p_{(n)}$ ) na qual este está inserido, considerando que quanto mais adiantada for a fase, maior o esforço para alteração.

$$w_{(n)} = c_{(n)} \times e_{(n)} \times P_{(n)}$$

Fórmula 6 – Cálculo do peso do artefato

Os níveis e valores relacionados às variáveis de complexidade, fase e esforço são determinados pelos desenvolvedores durante o processo de desenvolvimento. A variável  $e_{(n)}$  é determinada pelo esforço em horas necessárias para desenvolver um artefato e é mensurada em homem-hora. A complexidade e a fase devem ser classificadas em diferentes níveis. Para determinar a variável  $c_{(n)}$ , pode ser estabelecido o conjunto de níveis de complexidade  $\{alta; média; baixa\}$  com valores correspondentes a  $\{1;0,6;0,3\}$ . A variável  $p_{(n)}$  pode ser representada por  $\{Requisitos; Análise; Projeto; Implementação; Teste\}$  e ter valores associados como  $\{1;2;3;4;5\}$ , respectivamente. A figura 10 apresenta o diagrama de classes, que servirá de exemplo para o cálculo do peso de um artefato.

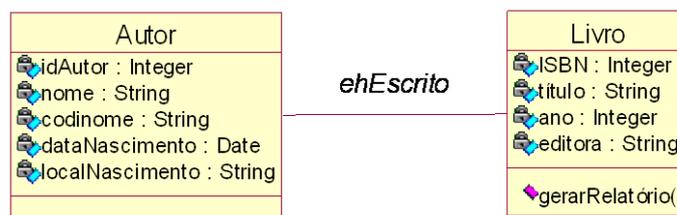


Figura 10 – Exemplo de diagrama de classes.

Usando os conjuntos de complexidade e fases acima propostos, pode-se determinar que o peso do artefato “Autor” é de  $w_{(a)}=0,45$ , considerando que o artefato tem complexidade baixa ( $c_{(a)}=0,3$ ), que o esforço de desenvolvimento é de 30 min ( $e_{(a)}=0,5$ ) e que está presente na fase de projeto ( $p_{(a)}=3$ ). Para o artefato “Livro”, o peso correspondente é de  $w_{(l)}=0,9$ , pois o seu tempo de desenvolvimento, incluindo implementação do método, é de 2h ( $e_{(l)}=2$ ), com complexidade baixa ( $c_{(l)}=0,3$ ) e também pertencente à fase de projeto ( $p_{(l)}=3$ ). Posteriormente, esses valores serão usados juntamente com a influência da relação entre os artefatos para determinar a métrica de impacto.

No âmbito da presente pesquisa, interessa descobrir quais artefatos serão afetados usando a probabilidade de impacto em cada um deles. Nesta realidade, a variável fase não é utilizada, uma vez que esta se refere as fases do processo de desenvolvimento e não a um projeto de manutenção. As variáveis de complexidade e esforço de cada artefato não determinam uma maior ou menor probabilidade de impacto, como é o objetivo, já que elas estão relacionadas à dificuldade de mantê-lo.

Utilizando o mesmo diagrama de classes da figura 10, com o uso do método Fast&&Serious [CAR02], pode-se determinar a complexidade das classes. A figura 11 mostra a aplicação do método para calcular a complexidade destas. Tem-se então que a complexidade da classe “Autor” é de  $CP_{(a)}=13$  e da classe “Livro” é de  $CP_{(l)}=17$ , sendo esta última mais complexa do que a primeira.

Ponto por Estado (SP)					
Classe	numLA	numHA	numIA	numAss	SP
Autor	5	0	0	1	5
Livro	4	0	0	1	4

Complexidade do Método (CM)				
Classe	Método	Triv/Sub	numIA	CM
Livro	Relatório	Sub	1	3

Pontos por Classe (CP)			
Classe	SP	BP	CP
Autor	5	1	13
Livro	4	3	17

Figura 11 – Aplicação do método Fast&&Serious.

Um exemplo de um requisito de mudança para o domínio desse problema poderia ser o apresentado na figura 12. O resultado dessa alteração no diagrama de classes do projeto possivelmente seria o representado na figura 13. Percebe-se então que, embora a classe “Livro” seja efetivamente mais complexa que a classe “Autor”, esse não foi um fator determinante para considerar o impacto do requisito de mudança.

Requisito de Mudança A: “*Deseja-se conhecer o grau de formação de cada autor. Tendo ele nível superior, informar qual a instituição de ensino onde obteve o grau.*”

Figura 12 – Requisito de mudança A, no domínio de livros.

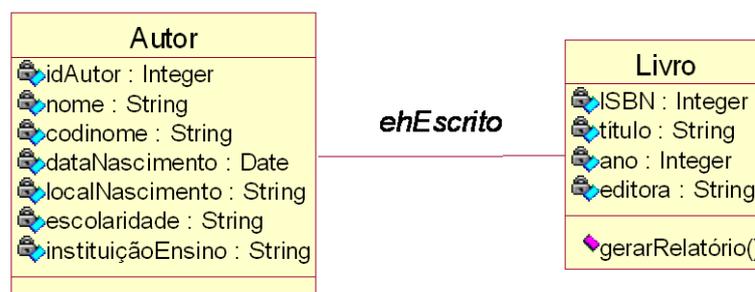


Figura 13 – Exemplo de diagrama de classes após alteração do requisito de mudança A.

### 3.4 INFLUÊNCIA DO CONCEITO DA ONTOLOGIA NOS ARTEFATOS

A metodologia TIAM utiliza fatores de influência nos elos que inter-relacionam o requisito de mudança e os diversos artefatos produzidos durante o projeto. Quem classifica a influência é o desenvolvedor, durante o processo de desenvolvimento. Os níveis de influência e seus respectivos valores podem ser escolhidos conforme a característica de cada projeto e pelo histórico de projetos da empresa. Uma vez escolhidos os níveis e valores atribuídos aos arcos entre os artefatos, o grau de influência deve ser normalizado, a fim de que, se vários artefatos estiverem relacionados a um único outro, a soma das influências de cada elo seja igual a 1, ou 100%. A figura 14 apresenta um exemplo do uso dessas influências.

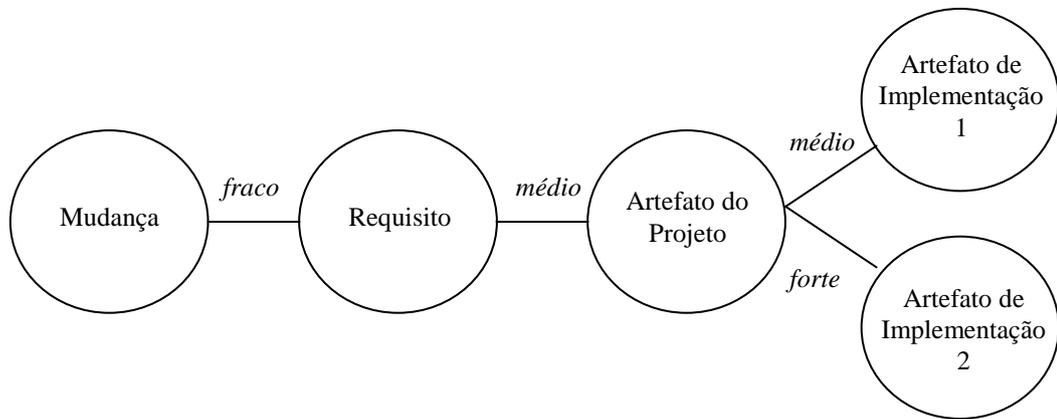


Figura 14 – Níveis de influência entre artefatos na metodologia TIAM.

O conjunto de níveis de influência da figura 14 é composto por  $\{fraco; médio; forte\}$ . Os valores associados a eles, sugeridos pelo autor, são  $\{0,3; 0,6; 1\}$ . Esses valores devem ser usados na fórmula 7, onde  $i(\{a,b\})$  é a influência entre dois artefatos,  $i'(\{a,b\})$  é o resultado da normalização e  $T$  é o conjunto de elos entre os artefatos. Na figura 15, é mostrado o resultado da influência normalizada de cada relacionamento.

$$i'(\{a,b\}) = \frac{i(\{a,b\})}{\sum_T i(\{c,b\})}$$

Fórmula 7 – Normalização dos elos na metodologia TIAM

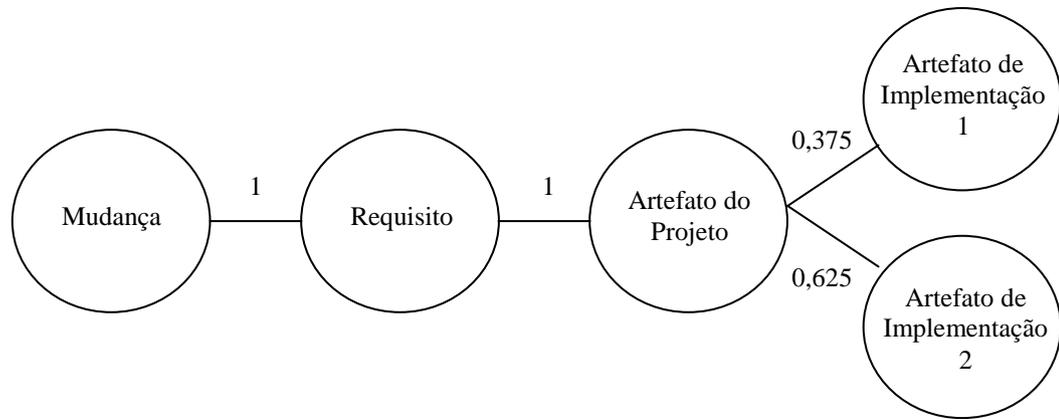


Figura 15 - Valores de influência normalizados entre artefatos na metodologia TIAM.

A metodologia TIAM proposta por O'Neal serve como base para a proposta da análise de impacto através da rastreabilidade por ontologias. Isso porque, considerando que uma ontologia tem representação equivalente a um grafo e que a rastreabilidade semântica nela relaciona conceitos a artefatos, pode-se assumir que existe uma influência específica do arco que liga o conceito ao artefato. A figura 16 representa graficamente essa relação, usando como exemplo os artefatos do grafo de relacionamento da figura 15.

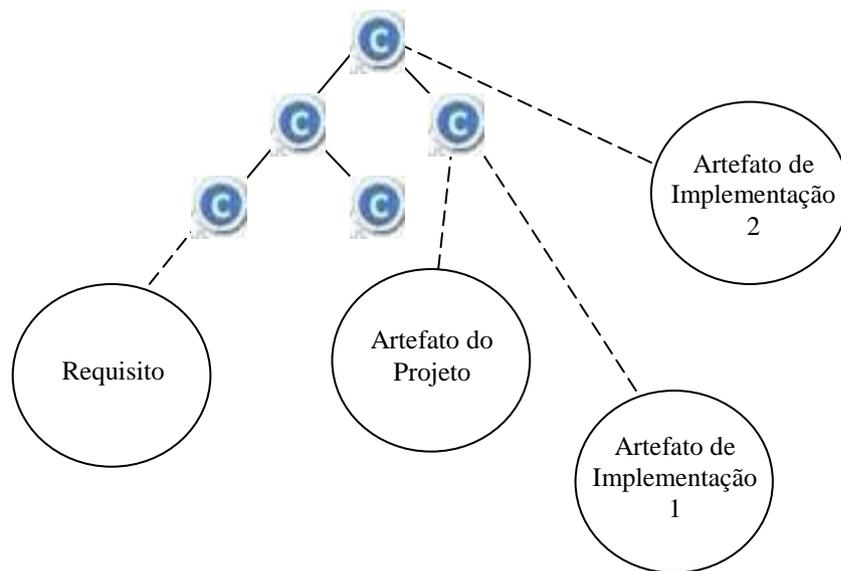


Figura 16 - Relacionamento entre conceitos da ontologia e artefatos de software.

O grau de influência de cada relacionamento depende do tipo de artefato ao qual o conceito está ligado. Embora o conjunto de níveis de influência e seus valores possam ser determinados por um desenvolvedor do projeto, baseado no histórico de dados deste, a metodologia aqui proposta sugere um conjunto inicial de níveis e de valores, obtidos de forma empírica. Estes são compostos por duas escalas  $\{fraco, forte\}$ , cujos valores associados são respectivamente  $\{0,3; 1\}$ .

Cada um dos conceitos, que estiver relacionado a um artefato, deve ter a variável *Influência do conceito no artefato (Ic)* informada, dentro da escala sugerida. A tabela V apresenta um *guideline* para ajudar nessa classificação, que é baseado em características do tipo do artefato.

Os diagramas da UML presentes na tabela V e que são referenciados nesse trabalho foram escolhidos, pois são suportados pela ferramenta *ArgoUML*, na qual o aplicativo ONTrace foi desenvolvido como um *plug-in*. Atualmente, só os objetos do ciclo de vida do software que foram modelados nesse sistema têm a possibilidade de serem rastreados.

Três dos diagramas que podem ser modelados e rastreados pela ferramenta de modelagem citada e que não estão presentes no *guideline* proposto são: o Diagrama de Casos de Uso, o Diagrama de Classes e o Diagrama de Implantação. O primeiro diagrama não consta na classificação da influência do conceito no artefato, pois se preferiu trabalhar somente com a descrição do caso de uso ao invés do relacionamento entre eles, uma vez que é no detalhamento deste que se encontram as informações comportamentais do sistema. De forma análoga, foram consideradas apenas as classes, pois assim é possível ter conhecimento exato de quais delas serão impactadas por um requisito de mudança, considerando que o Diagrama de Classes é muito abrangente, pois vários conceitos podem ser relacionados a ele. Já o último diagrama relatado não está abordado no *guideline* por dar a visão organizacional do *hardware* do sistema, que está fora do escopo dessa pesquisa.

Tabela V – *Guideline* para classificação da Influência do conceito no artefato (*Ic*)

Tipo do Artefato	Influência do conceito no artefato ( <i>Ic</i> )	
	Fraco	Forte
Descrição de Caso de Uso (UC)	O conceito é referenciado na pré ou pós-condições e/ou fluxos alternativo e de exceção	O conceito é referenciado no fluxo principal
Classe (C)	Classe associativa quando uma das classes ligadas a ela tem o mesmo nome do conceito	Possui o mesmo nome do conceito
Diagrama de Seqüência (DS)	Está presente nas mensagens entre os objetos	O conceito é um objeto
Diagrama de Estados (DE)	O conceito está presente nos estados dos fluxos alternativos e de exceções	O conceito está presente nos estados do fluxo básico
Diagrama de Atividades (DA)	O conceito está presente nas atividades dos fluxos alternativos e de exceções	O conceito está presente nas atividades do fluxo básico
Diagrama de Colaboração (DC)	Está presente nas mensagens entre os objetos	O conceito é um objeto

Os critérios de influência dos artefatos utilizados para construção do *guideline* foram determinados segundo estudo de cada um dos diagramas da UML. Por exemplo, quando



desenvolvimento do sistema e são atualizados durante o projeto para indicar a arquitetura física pretendida.

É importante diferenciar o tipo de diagrama, pois, dependendo do tipo de mudança, os artefatos são influenciados de forma diferente, com relação à sua classificação. Quando há alteração de regras de negócio, os diagramas dinâmicos são os mais afetados. Quando a mudança é referente à estrutura dos artefatos, os diagramas estáticos têm maior probabilidade de impacto. No experimento que foi realizado e que serviu para identificação das hipóteses, não foi possível detectar a influência nos diagramas funcionais, pois esses não fizeram parte do estudo. Sendo assim, a influência referente a esse tipo de diagrama será determinada como irrelevante, para qualquer tipo de mudança, sendo proposto seu estudo como trabalho futuro.

A *Influência do tipo de diagrama (Id)* é dada pelo conjunto das classificações dos diagramas UML {estático; dinâmico; funcional}. O peso para cada um desses elementos é dado conforme o tipo de mudança e será determinado na seção seguinte. Para cada artefato associado a um conceito, deve-se classificar o tipo de diagrama a que pertence, conforme a tabela VI.

Tabela VI – Classificação do tipo de diagrama

Tipo do Artefato	Influência do tipo de diagrama ( <i>Id</i> )		
	Estático	Dinâmico	Funcional
Diagrama de Caso de Uso (DUC)			X
Descrição de Caso de Uso (UC)		X	
Diagrama de Classes (DC)	X		
Classe (C)	X		
Diagrama de Seqüência (DS)		X	
Diagrama de Estados (DE)		X	
Diagrama de Atividades (DA)		X	
Diagrama de Colaboração (DCo)		X	
Diagrama de Implantação (DI)	X		

### 3.6 INFLUÊNCIA DO TIPO DE MUDANÇA NA ANÁLISE DE IMPACTO

Um requisito de mudança pode afetar de diferentes formas o produto ao qual é relacionado durante o projeto de manutenção deste. Considerando-se que durante o processo de desenvolvimento foi incluída a disciplina de Engenharia Ontológica, as alterações requisitadas podem ser identificadas na ontologia. Através do experimento, detalhado no Apêndice A, percebeu-se ser de grande influência as mudanças de regra de negócio, que estão relacionadas ao

significado dos conceitos da ontologia, e as mudanças no relacionamento dos conceitos, que engloba a inclusão ou alteração destes, alterando estruturalmente a ontologia.

A variável *Influência da mudança* ( $Im$ ) deve ser determinada para, juntamente com a *Influência do conceito no artefato* ( $Ic$ ) e a *Influência do tipo de diagrama* ( $Id$ ), medir o *Impacto no artefato pela mudança* ( $Iam_{(x,y)}$ ), que é calculado através da fórmula 8, onde  $x$  é o conceito do tipo *class* e  $y$  é o artefato.

$$Iam_{(x,y)} = Im \times Ic \times Id$$

Fórmula 8 – Cálculo do Impacto no artefato pela mudança ( $Iam_{(x,y)}$ )

A variável *Influência da mudança* ( $Im$ ) é formada pelo conjunto  $\{baixa; alta\}$ , com valores correspondentes a  $\{0,1;0,9\}$ , respectivamente. Os tipos de mudança detalhados a seguir possuem  $Im = alta$ . Terá  $Im = baixa$  qualquer outro tipo de mudança não citada. Uma vez que a influência do tipo de diagrama depende do tipo de mudança ocorrida, os valores dos elementos do conjunto  $\{estático; dinâmico; funcional\}$  serão determinados em cada uma delas.

### 3.6.1 Tipo de Mudança de Regras de Negócio

Quando uma regra de negócio já existente é alterada ou quando uma nova regra é associada a um conceito, o significado deste pode ser alterado. Isso significa que poderá haver mais restrições nas propriedades da ontologia ou que o domínio do conceito poderá ser alterado, por exemplo.

No código *Web Ontology Language* (OWL) presente na figura 18, tem-se a restrição *allValuesFrom*, que restringe os valores da propriedade da classe a que está associada. Ou seja, todos os membros da classe que possuírem a propriedade devem pertencer ao tipo de recurso indicado na cláusula. No exemplo significa que toda instância da classe “Casal”, se tiver filhos, estes devem pertencer a classe “filhoNatural”. Um requisito de mudança poderia alterar a obrigatoriedade de que os filhos do casal pertençam a classe de “filhoNatural”, podendo ser “filhosAdotivos”.

```

<owl:Class rdf:ID="Casal">
  <owl:Restriction>
    <owl:onProperty rdf:resource="#temFilhos"/>
    <owl:allValuesFrom rdf:resource="&filhoNatural"/>
  </owl:Restriction>
</owl:Class>

```

Figura 18 – Exemplo em OWL da restrição *allValuesFrom*

A influência dada por uma mudança de regra de negócio afeta diretamente os artefatos do tipo dinâmico, uma vez que esses detalham o comportamento do sistema. Sendo assim, os valores referentes para o conjunto  $Id = \{estático; dinâmico; funcional\}$  são, respectivamente,  $\{0,5; 1; 0,1\}$ .

### 3.6.2 Tipo de Mudança Estrutural

Quando conceitos são incluídos ou alterados numa ontologia, o relacionamento entre eles pode ser afetado. Numa inclusão, provavelmente um novo elo de ligação entre conceitos será necessário. Numa alteração, a própria ligação entre eles pode ser modificada. Nessas situações se está alterando estruturalmente o relacionamento entre os conceitos. Assim, o modelo de artefatos que sofre mais influência desse tipo de mudança são os diagramas estáticos, pois esses representam a estrutura do sistema. Desta forma, o conjunto dos valores referentes para o conjunto  $Id = \{estático; dinâmico; funcional\}$  são, respectivamente,  $\{1; 0,5; 0,1\}$ . As possibilidades de alteração na ontologia serão descritas abaixo.

#### 3.6.2.1 Alteração de um conceito da ontologia.

Quando um conceito é alterado, está se modificando a propriedade do tipo *range* de um conceito do tipo *datatypeProperty*. Essa mudança tem grande influência sobre artefatos relacionados diretamente a esse conceito. A figura 19 mostra uma sub ontologia para o domínio de livros, destacando o relacionamento do conceito “ISBN” com o seu tipo de dado.

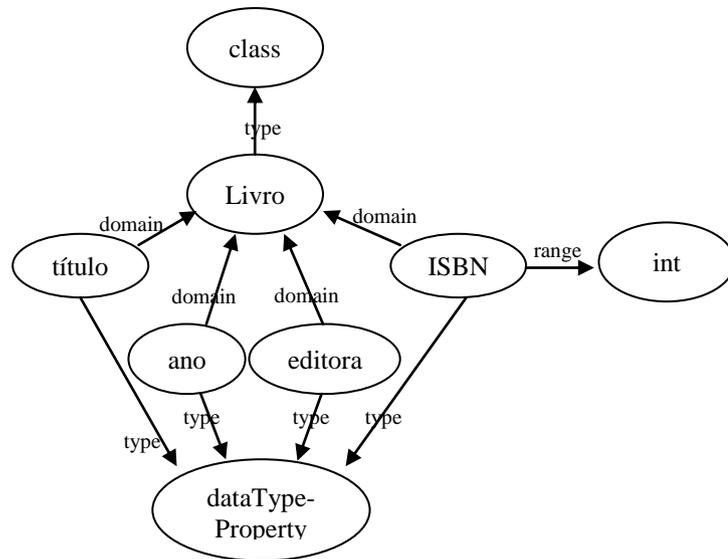


Figura 19 – Sub ontologia equivalente ao domínio de livros.

Na solicitação de mudança presente na figura 20, o *range* do conceito “ISBN” deve passar a ser “*String*”, como mostra a figura 21, referente à ontologia de mudança. Nesse contexto, são considerados os artefatos que estão ligados diretamente ao conceito do tipo *dataTypeProperty*, no exemplo, os artefatos ligados ao conceito “ISBN”.

Requisito de Mudança B: “*Através de um decreto do Ministério da Cultura, os livros editados no Brasil passarão a receber a sigla do país (BR) na identificação ISBN.*”

Figura 20 – Requisito de mudança B no domínio de livros.

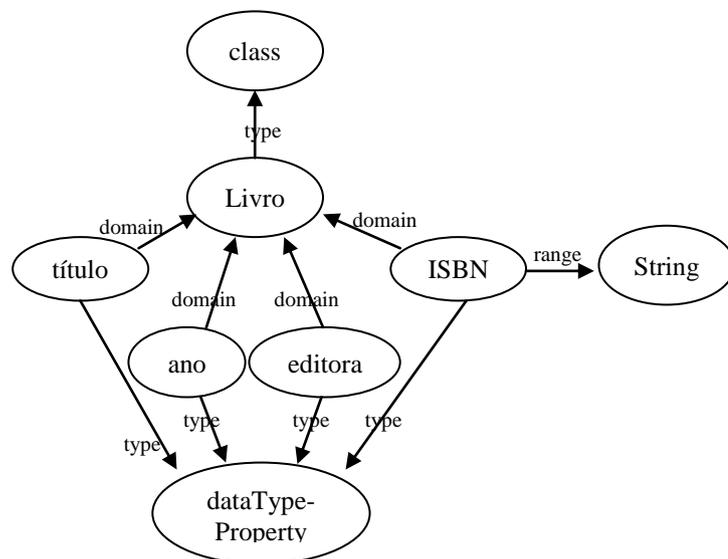


Figura 21 – Sub ontologia equivalente ao requisito de mudança no domínio de livros.

### 3.6.2.2 Inclusão de um conceito na ontologia

A inclusão de um conceito pode ser de dois tipos: a associação de um novo conceito do tipo *dataTypeProperty* a um já existente do tipo *class*, e a inclusão de um conceito do tipo *class*, com seus *dataTypeProperty* associados e *objectProperty* relacionados. A seguir, cada uma dessas inclusões será detalhada.

#### 3.6.2.2.1 Inclusão de conceito do tipo *dataTypeProperty*

A inclusão de um conceito do tipo *dataTypeProperty* irá acrescentar uma nova propriedade ao conceito do tipo *class* ao qual se relacionar. Essa situação pode ser verificada analisando o exemplo da figura 22 que é a representação de parte do modelo de domínio de livros, apresentado anteriormente na figura 10. A ontologia equivalente a esse modelo é mostrada na figura 23.

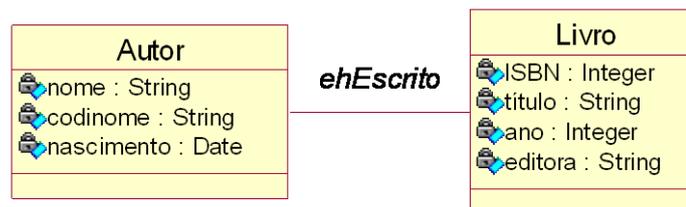


Figura 22 – Parte de um modelo de domínio de livros.

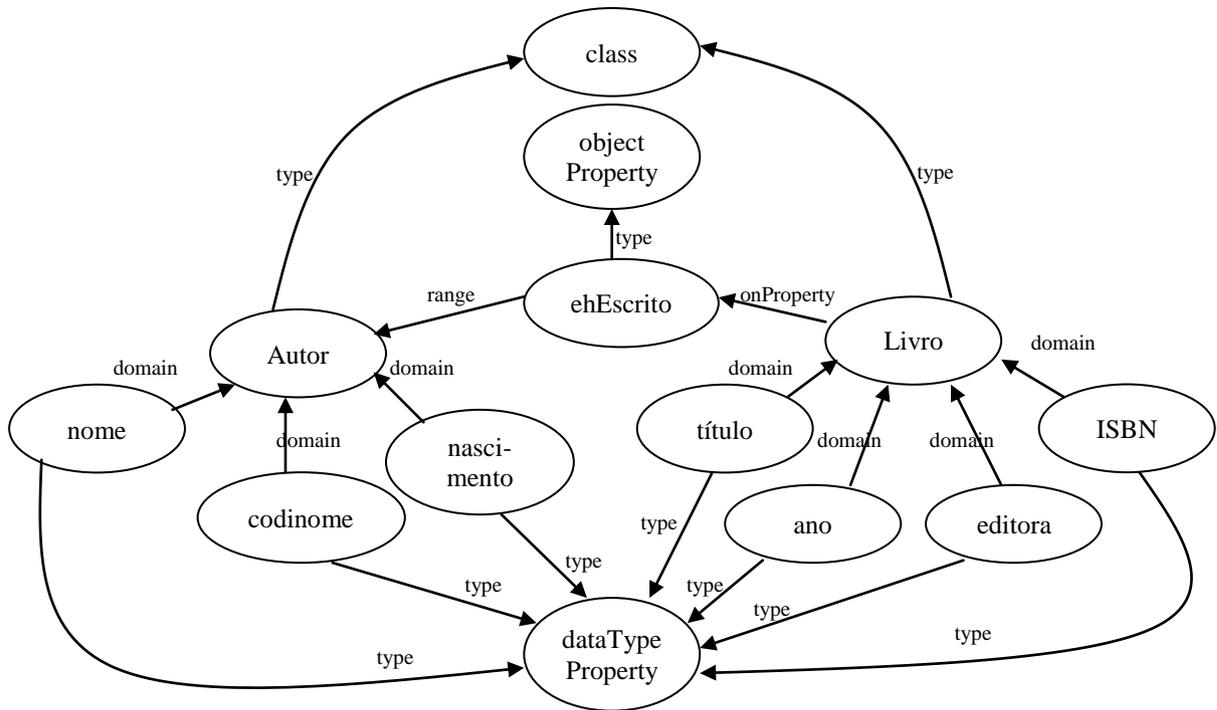


Figura 23 – Ontologia equivalente a parte de um modelo de domínio de livros.

Considerando o requisito de mudança da figura 12, a ontologia equivalente à mudança é mostrada da figura 24. Nota-se que um novo conceito “*formação*” do tipo *dataTypeProperty* foi associado ao conceito “*Autor*”, que é do tipo *class*.

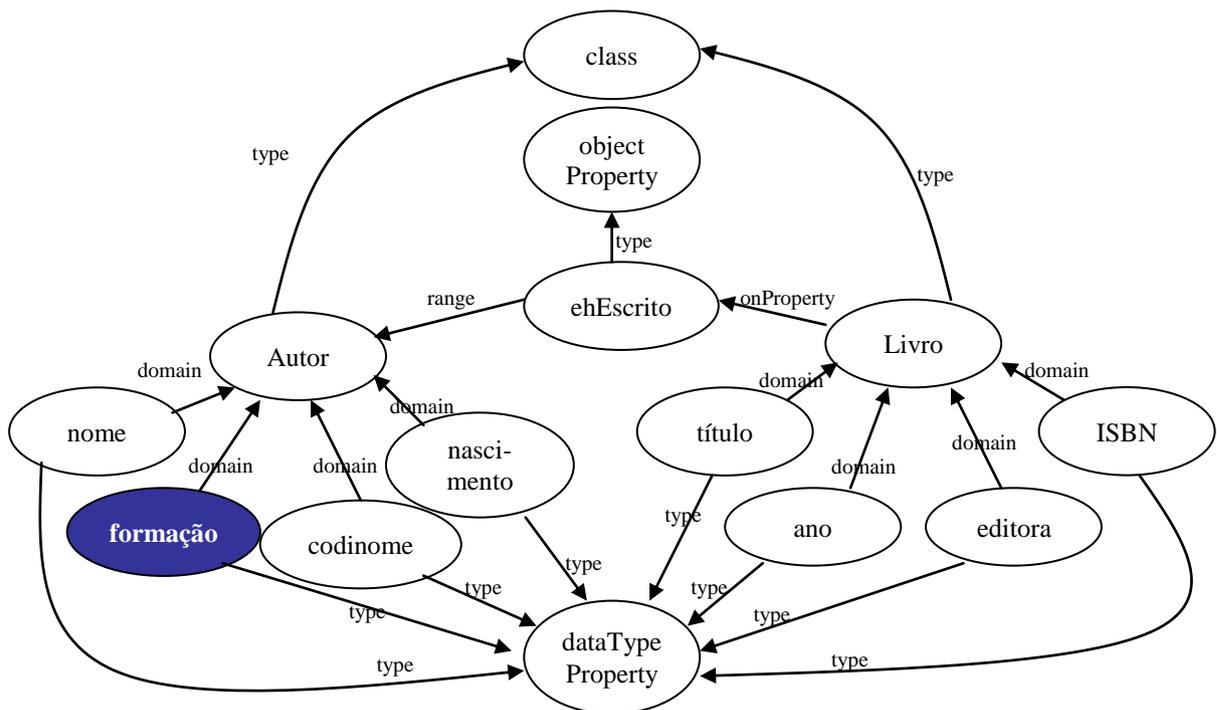


Figura 24 – Ontologia equivalente ao requisito de mudança A no domínio de livros.

A influência desse tipo de mudança foi percebida durante o experimento relatado no Apêndice A. A inclusão de um novo conceito do tipo *dataTypeProperty* caracterizou-se

por afetar fortemente os artefatos ligados ao conceito do tipo *class* ao qual o primeiro se relaciona.

### 3.6.2.2.2 Inclusão de conceito do tipo *class*

Durante o experimento, não houve nenhum requisito de mudança que propiciasse a inclusão de um novo conceito do tipo *class*. Estudando a estrutura ontológica percebe-se que quando conceitos desse tipo são incluídos, há uma nova associação com outros conceitos da ontologia. Essa associação é possibilitada pelo uso de conceitos do tipo *objectProperty*. Nesse caso, o impacto é calculado para os artefatos que estão relacionados ao conceito ao qual o novo conceito se liga.

Para exemplificar essa situação, o requisito de mudança da figura 25 é aplicado ao modelo de domínio de livros, representado na figura 22. A alteração na ontologia está representada na figura 26, onde estão destacados os novos conceitos incluídos.

Requisito de Mudança C: “É necessário conhecer em quais livrarias o livro está disponível para venda. As informações necessárias sobre esta são: onde está localizada, qual o volume de vendas por mês, contato e ramo de atividade.”

Figura 25 – Requisito de mudança C no domínio de livros.

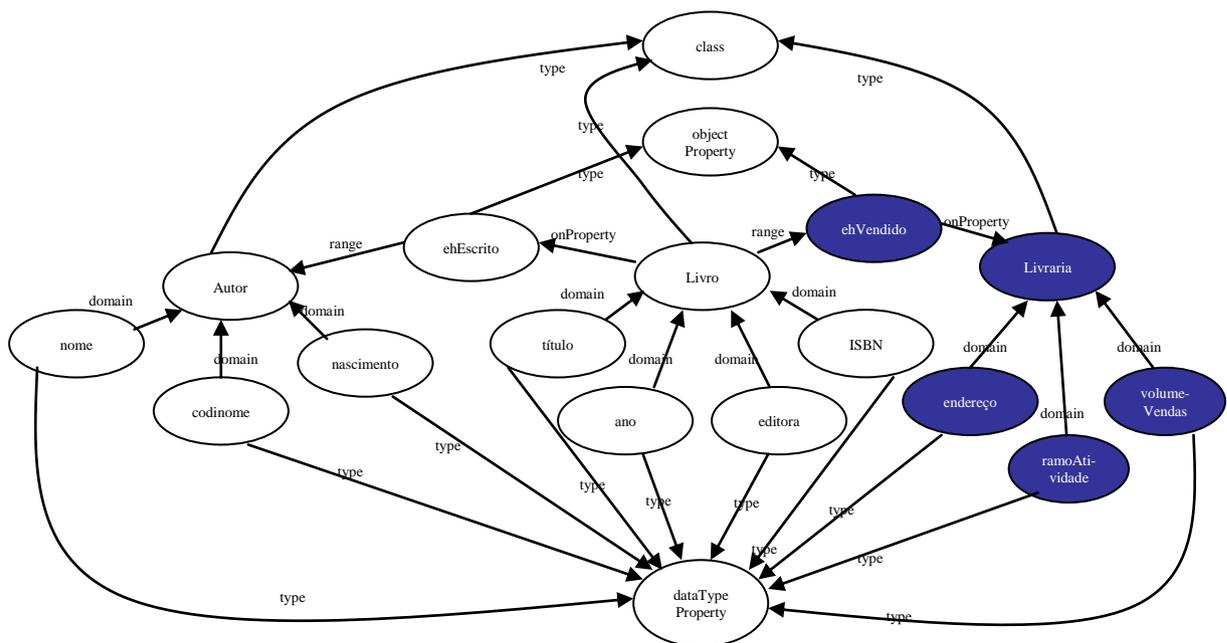


Figura 26 – Ontologia equivalente ao requisito de mudança C no domínio de livros.

No exemplo, o novo conceito do tipo *class* “Livraria” é ligado ao conceito do tipo *class* “Livro”, anteriormente existente, através do novo conceito do tipo *objectProperty* “eh-Vendido”. Como na ontologia original do domínio de livros não existe o conceito “Livraria”, conseqüentemente não há artefatos rastreados para esse conceito. Assim, a análise de impacto dessa mudança deve ser feita rastreando-se os artefatos que estão associados ao conceito “Livro”, através da influência do relacionamento entre conceitos, apresentado a seguir.

### 3.7 INFLUÊNCIA DO RELACIONAMENTO ENTRE CONCEITOS

Sendo a ontologia equivalente a um grafo, é possível percorrer seus nodos, ou conceitos, a partir de critérios pré-estabelecidos. No contexto da rastreabilidade semântica, essa característica é útil, pois permite encontrar os artefatos indiretamente afetados, a partir do relacionamento entre conceitos.

A metodologia aqui proposta sugere o uso de algoritmos de radicalização (ou de *stemming*) para língua portuguesa, como forma de identificar a relação sintática existente entre os conceitos que se relacionam. Este é utilizado para determinar o peso relativo ao conceito do tipo *objectProperty* que liga outros dois conceitos do tipo *class*. Na seção seguinte, será abordada a área na qual esse método está inserido, bem como a forma de utilizá-lo.

#### 3.7.1 Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN) é usado, segundo Dias [DIA04], para descrever a função de softwares ou de componentes de hardware em um sistema computacional que analisam ou sintetizam linguagem falada ou escrita. Segundo Gonzalez [GON03], aspectos da comunicação humana como sons, palavras, sentenças e discursos, considerando formatos e referências, estruturas e significados, contextos e usos são tratados computacionalmente. O autor estabelece níveis de entendimento para esses aspectos:

- a) Fonético e fonológico: que trabalham o relacionamento das palavras com os seus sons;

- b) Morfológico: abrange a construção das palavras a partir das unidades de significado primitivas e sua classificação em categorias morfológicas;
- c) Sintático: aborda o relacionamento das palavras entre si e como as frases podem ser partes de outras, construindo sentenças;
- d) Semântico: estuda o relacionamento das palavras com seus significados e como eles são combinados para formar os significados das sentenças;
- e) Pragmático: abrange o uso de frases e sentenças em diferentes contextos, afetando o significado.

No contexto da presente proposta, cabe analisar os níveis morfossintático e semântico. Ambos serão brevemente detalhados a seguir, citando as técnicas utilizadas para extrair informações de cada um desses processamentos.

### 3.7.1.1 Processamento Morfossintático

Segundo Dias [DIA04], fazem parte do processamento morfossintático, a análise morfológica e a análise sintática. Ambas preocupam-se com a constituição das palavras e de seus grupos, que é o que forma os elementos de expressão de um língua. A morfologia está ligada a estrutura da palavra e a classificação dessas. Por outro lado, a análise sintática preocupa-se com o agrupamento de palavras, verificando a formação de frases.

Inserido na análise morfológica encontra-se a conflação, que é o ato de fusão ou combinação para igualar variantes morfológicas de palavras. Segundo Gonzalez [GON03] os principais métodos de conflação são o de radicalização e o de lematização.

O método de radicalização é relevante, dentro da proposta de análise de impacto, pois combina as formas diferentes de uma palavra em uma representação comum, o radical. Esse método pode ser utilizado para determinar a similaridade entre os conceitos da ontologia, através de algoritmos de radicalização. Esses algoritmos permitem comparar palavras que são basicamente iguais, mas que sem a radicalização são palavras distintas. Na figura 27, é mostrado um exemplo de aplicação dessa técnica. As palavras “Livro” e “Livraria” são reduzidas a um mesmo radical, identificando que há uma relação entre elas.

<p><b>Livro</b>  <b>Livraria</b>  Radical: <i>livr-</i></p>
---

Figura 27 – Exemplo de radicalização de palavras.

Na metodologia de análise de impacto, se for possível reduzir os conceitos ao mesmo radical, então a influência do relacionamento entre esses dois conceitos é alto, caso contrário é baixo. Os níveis de *Influência do relacionamento* ( $Ir$ ) podem então ser classificados em:  $\{baixo; alto\}$ , tendo como valores correspondentes  $\{0,5; 0,9\}$ .

A ligação entre os conceitos “Livro” e “Livraria” é dada pelo conceito “ehVendido”, como visto na figura 28, onde estão destacados os conceitos do tipo *objectProperty*. Nesse caso, a *Influência do relacionamento* seria:  $Ir = alto$ . Isso significa que um requisito de mudança que afete os artefatos ligados a um desses conceitos tem grande probabilidade de impactar os artefatos do outro. Já o conceito “ehEscrito” tem influência  $Ir = baixo$ , pois os conceitos relacionados a “Autor” e “Livro” não podem ser reduzidos ao mesmo radical.

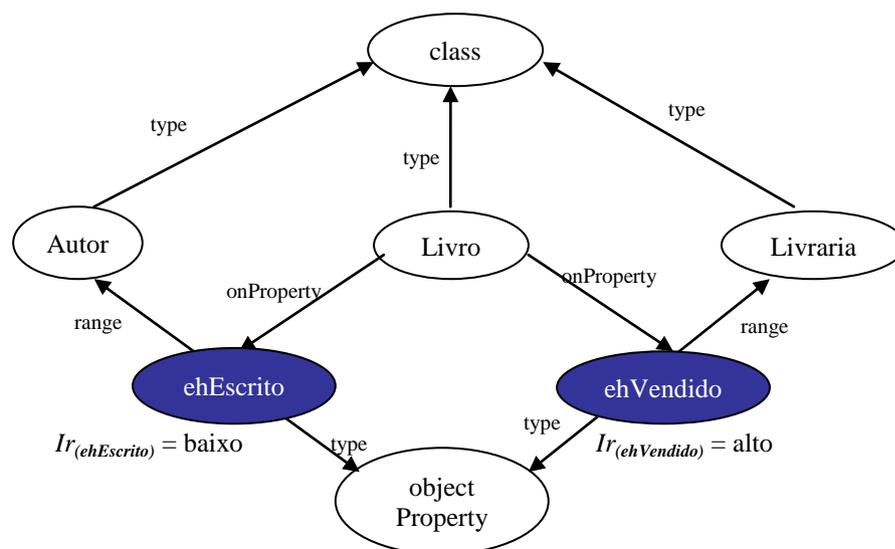


Figura 28 – Sub ontologia equivalente ao domínio de livros, destacando a propriedade *objectProperty*.

Embora a extração de um radical comum entre dois termos seja capaz de estabelecer um relacionamento entre eles, este está restrito à morfologia das palavras. Uma maneira mais precisa determinar a influência de um relacionamento seria usar estruturas semânticas para tal. Na seção seguinte será brevemente detalhada a estrutura de tesauro.

### 3.7.1.2 Processamento Semântico

A semântica tem relação com o significado das palavras ou do conjunto delas. Segundo Dias [DIA04], o processamento semântico é considerado um dos grandes desafios do

PLN, pois se vincula a morfologia e a estrutura sintática juntamente com informações pragmáticas.

O significado de conjuntos de palavras pode ser obtido utilizando estruturas como um tesouro, cuja definição dada por Corrêa [COR03] é de um agrupamento de palavras, ou radicais de palavras em certas categorias de assunto chamadas de “classes conceituais”. Uma das vantagens de usá-lo é a possibilidade de se ter um vocabulário controlado para pesquisa.

Os objetivos elencados por Salton [SAL68] para a criação de um tesouro são:

- a) A padronização das palavras-chave que foram encontradas, ou seja, fornecer um vocabulário padrão para indexação e pesquisa;
- b) Ajudar os usuários na localização de termos para a formulação de consultas e;
- c) Fornecer hierarquias de classes que permitem ampliar e limitar as consultas de acordo com as necessidades do usuário.

A forma mais simples de estruturar um tesouro é ter uma lista de palavras (conceitos) importantes para um domínio de conhecimento e, para cada palavra dessa lista, um conjunto de palavras relacionadas. As palavras relacionadas são variações derivadas de um relacionamento sinônimo.

A representação, segundo Corrêa [COR03], pode ser feita através de um grafo, onde cada nodo representa um termo que está ligado a outros termos e aos arcos são atribuídos pesos. Os pesos são importantes para poder determinar a similaridade entre termos de um documento. A figura 29 representa um tesouro.

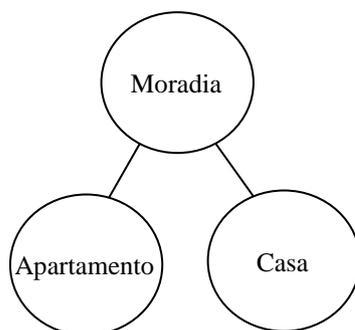


Figura 29 – Exemplo de tesouro.

Em um tesouro palavras sintaticamente distintas, mas com semântica equivalente podem ser relacionadas. No exemplo da figura 29, os termos “casa” e “apartamento” têm significados iguais, no contexto de moradia, mas são sintaticamente diferentes.

No contexto de análise de impacto, uma melhoria na proposta aqui descrita seria usar estrutura de tesouro, ao invés de algoritmos de radicalização, para determinar a influência do relacionamento entre conceitos da ontologia. Assim, termos relacionados a um mesmo

domínio poderiam ser identificados, mesmo que tenham formação morfológica distintas. Dessa forma, a precisão da influência entre os conceitos seria maior, possibilitando determinar mais precisamente a probabilidade de impacto de uma mudança em diversos níveis de caminhamento na ontologia.

A seguir, será apresentado o cálculo da influência do relacionamento entre conceitos, utilizando o processo de radicalização de palavras, como visto anteriormente. O uso de uma estrutura como tesauro deverá ser verificada em trabalhos futuros.

### 3.7.2 Cálculo da Influência do Relacionamento entre Conceitos

Na ontologia, a propriedade *objectProperty* tem o papel de relacionar conceitos do tipo *class*. A figura 28 destaca essa propriedade. Assim, podemos perceber que, se uma alteração for realizada no conceito “Autor”, os artefatos ligados ao conceito “Livro” podem ser rastreados, uma vez que os dois estão relacionados através do conceito “ehEscrito”. Se ocorrer uma mudança em “Livro”, essa pode afetar os artefatos relacionados ao conceito “Livraria”, já que esses estão ligados pelo conceito “ehVendido”. Assim, a rastreabilidade pode ser realizada em níveis. Na metodologia de análise de impacto aqui proposta, o rastreamento de conceitos é realizado somente na ontologia do sistema após os conceitos do requisito de mudança serem identificados nela. No capítulo 4, esse assunto será amplamente abordado.

Associado a cada conceito do tipo *objectProperty* tem-se a influência que este exerce sob os conceitos aos quais se relaciona. Esta influência é importante uma vez que irá auxiliar na determinação de em quantos níveis é interessante percorrer a ontologia. Se não houver um critério para determinar o nível máximo de rastreabilidade, é provável que todos os artefatos do projeto sejam encontrados, uma vez que todos os conceitos estão relacionados de alguma forma. Para iniciar o caminhamento na ontologia, sugere-se rastrear a partir do conceito que teve maior número de mudanças, seja estrutural ou semântica.

Deve-se encontrar o conceito onde se iniciará o percurso de busca, somando todos os valores de *Impacto no artefato pela mudança* ( $Iam_{(x,y)}$ ) que foram encontrados para cada um dos artefatos relacionados aos conceito. Aquele conceito que possuir o maior *Impacto acumulado* ( $Ia_{(x)}$ ), será o ponto inicial de caminhamento no grafo, apresentado na fórmula 9 onde  $x$  é o conceito,  $n$  é o número de artefatos ligados ao conceito e  $y$  é o artefato.

$$Ia_{(x)} = \sum_{x=1}^n Iam_{(x,y)}$$

Fórmula 9 – Cálculo do Impacto acumulado ( $Ia_{(x)}$ )

A figura 30 mostra a aplicação da fórmula 9, supondo valores para artefatos ligados aos conceitos “Livraria”, “Livro” e “Autor”. Com o resultado, o conceito a partir do qual se iniciará o caminhamento é o conceito “Livraria”.

$Iam_{(x,y)} = Im \times Ic \times Id$
$Iam_{(Autor, C-Autor)} = 0,9 \times 1 \times 0,5 = 0,45$
$Iam_{(Autor, UC-ManterAutor)} = 0,9 \times 1 \times 1 = 0,9$
$Ia_{(Autor)} = 1,35$
$Iam_{(Livro, C-Livro)} = 0,9 \times 1 \times 0,5 = 0,45$
$Iam_{(Livro, DA-ManterLivro)} = 0,9 \times 1 \times 1 = 0,9$
$Ia_{(Livro)} = 1,35$
$Iam_{(Livraria, UC-ManterLivraria)} = 0,9 \times 1 \times 1 = 0,9$
$Iam_{(Livraria, DA-ClassificarLivraria)} = 0,9 \times 1 \times 1 = 0,9$
$Ia_{(Livraria)} = 1,8$

Figura 30 – Exemplo de cálculo do Impacto acumulado ( $Ia_{(x)}$ ).

Para estabelecer a *Propagação do impacto* ( $Pi$ ), os diversos fatores de *Influência do relacionamento* ( $Ir_{(z)}$ ) devem ser multiplicados, como mostra a fórmula 10, onde  $z$  representa o conceito do tipo *objectProperty*, conforme os conceitos são percorridos. A sugestão dessa a metodologia é que o nível de impacto para parada do percurso seja fixado em 70%. Sendo menor do que esse valor não é mais recomendável continuar a rastreabilidade, pois a tendência é que os artefatos não sejam afetados pela alteração do requisito de mudança.

$$Pi = Ir_{z1} \times Ir_{z2} \times \dots \times Ir_{zn}$$

Fórmula 10 – Cálculo da Propagação do impacto ( $Pi$ )

Todos os conceitos do tipo *objectProperty* que estiverem relacionados ao conceito com maior  $Ia_{(x)}$  devem ser analisados para que o *Impacto no artefato pelo relacionamento* ( $Iar_{(x,y)}$ ) seja calculado. Essa variável irá determinar a probabilidade de alteração no artefato, através da fórmula 11, onde  $x$  é o conceito do tipo *class* e  $y$  é o artefato. Nela,  $Pi$  representa a propagação de influência que o conceito recebeu até o momento no caminho percorrido no grafo. Os valores que devem ser atribuídos aos elementos da variável  $Id$  são os mesmos utilizados para mudanças de regras de negócio, isso porque supostamente as alterações desse tipo têm maior propagação do que mudanças estruturais.

$$Iar_{(x,y)} = Pi \times Ic \times Id$$

Fórmula 11 – Cálculo do Impacto no artefato pelo relacionamento ( $Iar_{(x,y)}$ ).

Usando os elementos da figura 28, tem-se que o conceito do tipo *class* “Livraria” está ligado ao conceito do tipo *objectProperty* “ehVendido” que por sua vez está ligado ao conceito do tipo *class* “Livro”. Considerando que os artefatos do conceito “Livraria” foram os que mais sofreram alterações, os artefatos do conceito “Livro” têm probabilidade de serem impactados. O calculo dessa probabilidade é mostrada na figura 31. Nota-se que  $Pi$  é igual a  $Ir_{(ehVendido)}$ , isso porque os conceitos estão diretamente ligados.

$$\begin{aligned}
 Iar_{(x,y)} &= Pi \times Ic \times Id \\
 Iar_{(Livro, C-Livro)} &= Pi \times Ic \times Id = 0,9 \times 1 \times 0,5 = 0,45 \\
 Iar_{(Livro, DA-ManterLivro)} &= 0,9 \times 0,9 \times 1 = 0,9
 \end{aligned}$$

Figura 31 – Exemplo do cálculo do Impacto no Artefato pelo Relacionamento ( $Iar_{(x,y)}$ )

Uma vez que o conceito “Livro” tem probabilidade de ser impactado e este se relaciona com o conceito do tipo *class* “Autor” através do conceito do tipo *objectProperty* “ehEscrito”, os artefatos relacionados ao conceito “Autor” podem ser também impactados pela mudança. Calcula-se então a *Propagação do impacto* ( $Pi$ ) e as probabilidades correspondentes para cada artefato, como mostra a figura 32. Dessa forma, estamos percorrendo a ontologia em três níveis.

$$\begin{aligned}
 Pi &= Ir_{(ehEscrito)} \times Ir_{(ehVendido)} \\
 Pi &= \textit{baixo} \times \textit{alto} \\
 Pi &= 0,5 \times 0,9 = \mathbf{0,45} \\
 \\ 
 Iar_{(x,y)} &= Pi \times Ic \times Id \\
 Iar_{(Autor, C-Autor)} &= 0,45 \times 0,9 \times 0,5 = 0,20 \\
 Iar_{(Autor, UC-ManterAutor)} &= 0,45 \times 0,9 \times 1 = 0,40
 \end{aligned}$$

Figura 32 – Exemplo de cálculo da Propagação do impacto ( $Pi$ ).

No exemplo, utilizou-se um critério de parada de 40% com o único objetivo de demonstrar o cálculo em níveis, mesmo sabendo que não é esse o fator recomendado pela metodologia aqui apresentada. Nesse caso específico, quando  $Pi < 0,4$ , o caminhamento no grafo deve ser descontinuado. Por exemplo, se o conceito “Autor” estivesse ligado a um outro conceito e o  $Ir$  entre eles fosse baixo, o valor de  $Pi$  passaria a ser de 0,225, tornando assim não interessante o rastreamentos dos artefatos ligados a esse último conceito.

Após o detalhamento de todas as influências e variáveis necessárias para que a metodologia seja aplicada, o próximo capítulo apresentará os passos que devem ser realizados para que a análise de impacto de um requisito de mudança seja estabelecida.

#### 4 METODOLOGIA PARA ANÁLISE DE IMPACTO DE MUDANÇA

A metodologia de análise de impacto baseada na rastreabilidade por ontologias avalia cada requisito de mudança de software separadamente. Esses requisitos são analisados no contexto de projetos de manutenção de um produto, sendo que devem ser referentes a alterações funcionais no software. Pela aplicação da metodologia, os artefatos com maior probabilidade de serem afetados pela mudança são identificados.

A metodologia de análise de impacto de mudanças baseada na rastreabilidade por ontologias é composta por quatro passos: gerar a ontologia do requisito de mudança, identificar os conceitos da ontologia de mudança na ontologia do sistema, rastrear os artefatos impactados e listar, ordenados decrescentemente, os artefatos impactados. A figura 33 mostra graficamente como esses passos são encadeados.

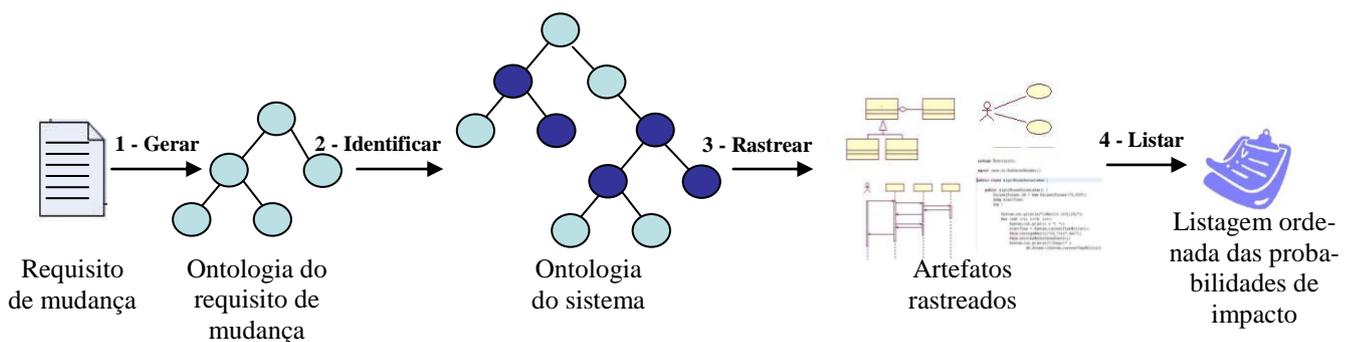


Figura 33 – Passos da metodologia de análise de impacto baseada em ontologias

#### 4.1 GERAR ONTOLOGIA DO REQUISITO DE MUDANÇA

O primeiro passo para gerar a ontologia do requisito de mudança é, a partir de um requisito de mudança, construir o modelo conceitual deste. Essa construção deve ser feita utilizando o dicionário de dados do sistema, que é definido como “um depósito central que descreve e define o significado de toda a informação usada na construção de um sistema”, dado por Oliveira [OLI01]. Este artefato tem como objetivo, uniformizar a nomenclatura dos termos que serão usados na modelagem do requisito de mudança. Por exemplo, se no sistema é usado o termo “docente”, não deve ser usado na modelagem do requisito de mudança o termo “professor”. A figura 34 mostra esse processo.

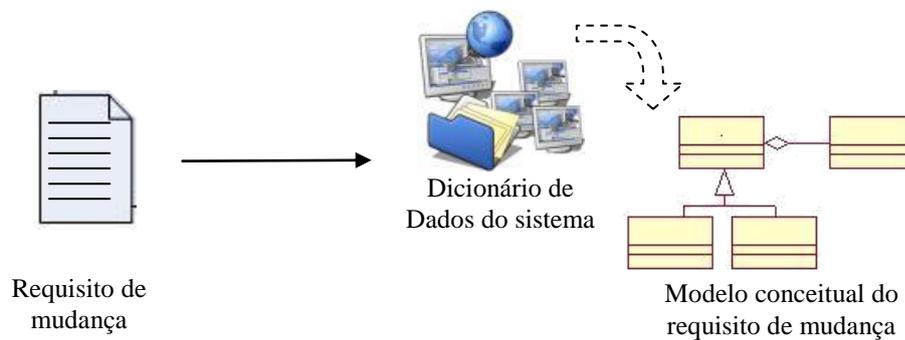


Figura 34 – Desenvolvimento do modelo conceitual do requisito de mudança

Após o modelo conceitual ter sido criado, pode-se gerar a ontologia referente a ele. Sugere-se o uso da ferramenta ONTrace da mesma forma como a ontologia do sistema é construída. A figura 35 apresenta esse passo.

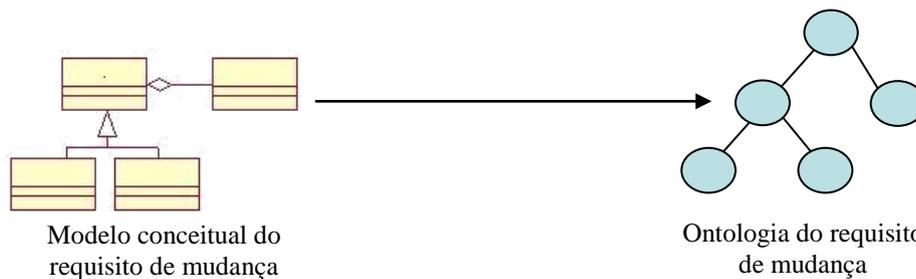


Figura 35 – Gerar ontologia do requisito de mudança a partir de um modelo conceitual

A ontologia do requisito de mudança contém os conceitos existentes na ontologia do sistema, sendo um sub-conjunto deste. A intersecção entre as ontologias é formada pelos conceitos que são comuns as duas e que serão analisados para identificar o impacto da mudança. Não é possível realizar a análise de impacto se os conjuntos forem disjuntos, pois nesse

caso, não havendo nenhum conceito em comum, não há como realizar a rastreabilidade dos artefatos, o que impede a medição do impacto.

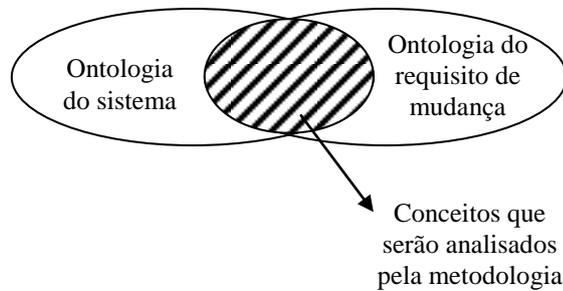


Figura 36 – Intersecção entre ontologias

A próxima fase da metodologia é responsável por identificar os conceitos da ontologia do requisito de mudança na ontologia do sistema, que formam o conjunto comum às duas, representado na figura 36.

## 4.2 IDENTIFICAR CONCEITOS NA ONTOLOGIA DO SISTEMA

A segunda fase da metodologia é identificar, na ontologia do sistema, os conceitos que foram modelados no requisito de mudança. Esse passo é necessário uma vez que a rastreabilidade entre conceito e artefato foi realizada na ontologia do sistema. A figura 37 apresenta esse passo.

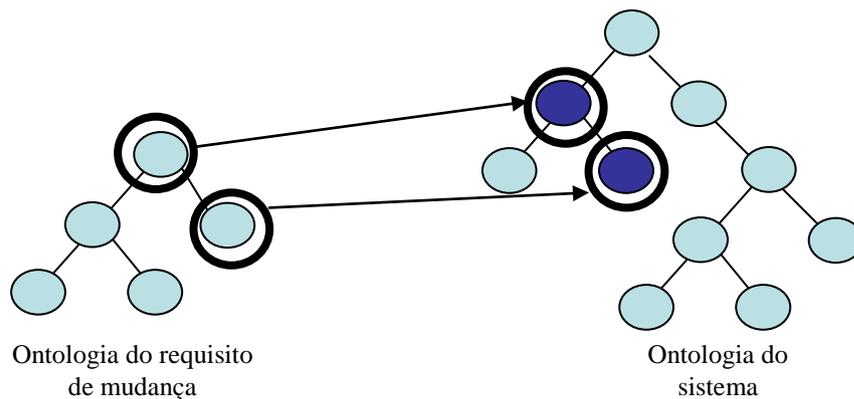


Figura 37 – Identificação dos conceitos da ontologia do requisito de mudança na ontologia do sistema

O processo de identificação deve ser realizado de forma a encontrar o tipo de mudança que está ocorrendo, explicitada na seção 3.6. Eventualmente, alguns conceitos da ontologia do requisito de mudança não serão encontrados na ontologia do sistema, o que caracteri-

za uma inclusão. Em outros casos, podem-se encontrar conceitos que estão sendo alterados, e, também, conceitos que foram apenas modelados sem sofrer nenhum tipo de alteração.

Na figura 38, tem-se a representação de um conceito do tipo *dataTypeProperty* sendo incluído, que é “taxaGordura”. Os conceitos “dataAvaliação” e “FichaBiométrica”, que estão destacados, foram identificados na ontologia do sistema.

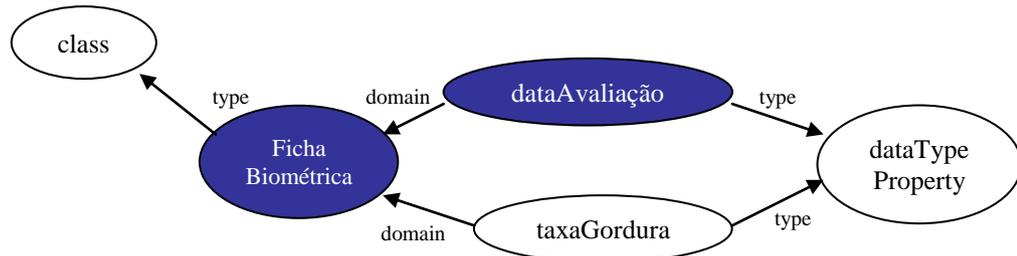


Figura 38 – Identificação de inclusão de um conceito do tipo *dataTypeProperty*

A figura 39 apresenta um conceito que está sendo alterado. Na ontologia referente ao requisito de mudança, o *range* do conceito “Meta”, em destaque, é “int”. Na ontologia do sistema, esse mesmo conceito tem *range* “string”.

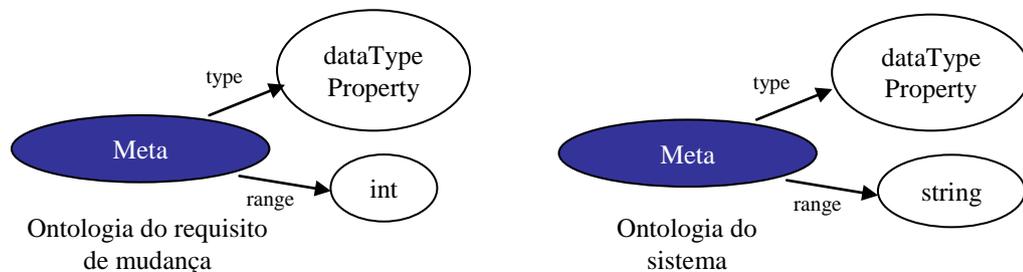


Figura 39 – Identificação de alteração de um conceito do tipo *dataTypeProperty*

Os conceitos que são identificados na ontologia do sistema, mas que não sofrem alterações estruturais, tais como inclusão ou alteração de conceitos, podem ter alterações semânticas, ou seja, uma regra de negócio associada ao conceito pode ser alterada ou incluída. Nesse caso, o responsável por fazer a análise da mudança deve indicar que o conceito é afetado dessa forma. Na figura 40, o conceito “Cliente” foi identificado na ontologia do sistema, e este pode ser afetado ou não por uma regra de negócio constante no requisito de mudança.

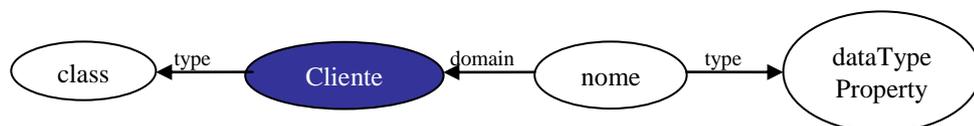


Figura 40 – Identificação de um conceito do tipo *dataTypeProperty* que não sofre mudança

A não existência de um conceito na ontologia do requisito de mudança, do tipo *class* ou *dataTypeProperty*, não caracteriza que esse conceito está sendo excluído da ontolo-

gia do sistema. Essa situação deverá ser tratada em trabalhos futuros por apresentar complexidade superior à inclusão ou alteração de conceitos.

Outra identificação que pode ser feita é quanto aos relacionamentos existentes entre conceitos. Esses relacionamentos permitem encontrar artefatos em  $n$ -níveis. O conceito da ontologia do sistema que mais sofreu mudanças, seja de inclusão de conceitos ou de alteração destes, será o ponto inicial de pesquisa, como visto no capítulo anterior. A partir desse ponto, o conceito do tipo *objectProperty* relacionado a ele é encontrado e inicia-se então o percurso pelo encadeamento de conceitos. Um conceito do tipo *objectProperty* é responsável por relacionar outros dois conceitos do tipo *class*. A figura 41 apresenta esse tipo de relacionamento, onde se pode determinar, como exemplo, que o conceito de partida seja o conceito “Mensalidade”, pois foi o mais afetado por mudanças.

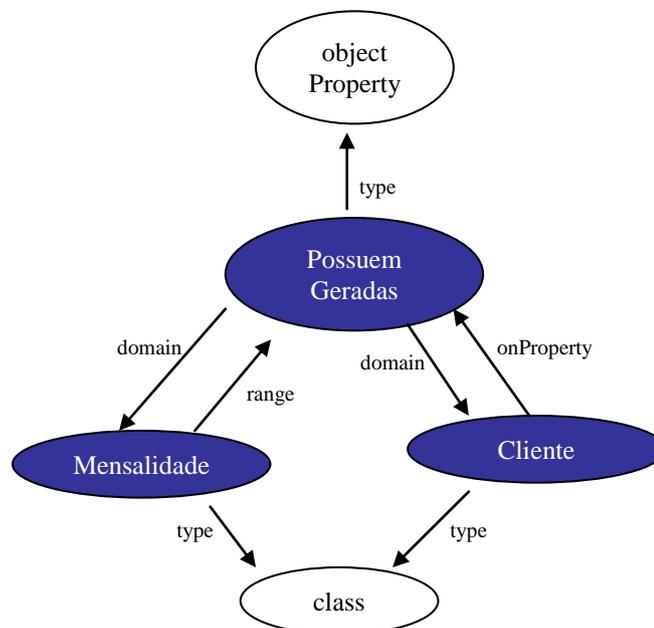


Figura 41 – Identificação de relacionamento entre conceitos

### 4.3 RASTREAR ARTEFATOS

Após a identificação dos conceitos na ontologia do sistema, os artefatos relacionados a este podem ser rastreados. Os artefatos são rastreados conforme a relação e o tipo de mudança, e a esta é atribuída uma *Influência da mudança (Im)*.

### 4.3.1 Relação Conceito-Artefato

Esse tipo de relacionamento é usado para rastrear os artefatos em apenas um nível, ou seja, que estão ligados diretamente aos conceitos. O cálculo do impacto deve ser realizado para cada um dos artefatos e é determinado pela fórmula 8.

A figura 42 abaixo mostra, genericamente, como é dada a relação entre um conceito extraído da ontologia do sistema e artefatos rastreados a partir dele. No exemplo, o conceito “Mensalidade” foi identificado como existente na ontologia do sistema. A esse conceito, foram ligados os artefatos do tipo caso de uso “UC-GerarMensalidade” e do tipo classe “C-Mensalidade”.

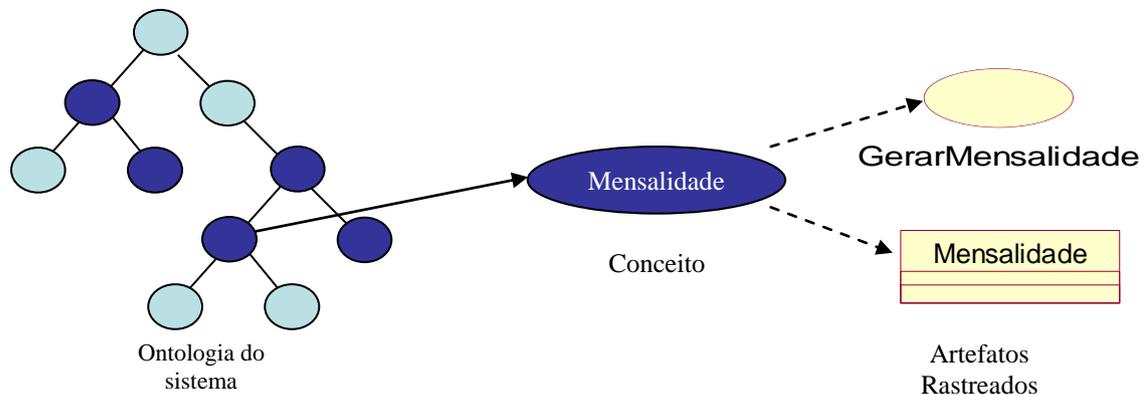


Figura 42 – Rastreabilidade de relação conceito-artefato

Para realizar a rastreabilidade nesse tipo de relação, é importante distinguir se a alteração é referente a regras de negócio ou a relacionamentos, o que muda estruturalmente a ontologia. Esses tipos de mudança têm influência alta, mas são diferenciados na maneira como o artefato é impactado, dependendo do tipo de diagrama da UML a que pertencem. Além disso, as alterações estruturais podem ser identificadas automaticamente, pois há a inclusão de um novo conceito ou a alteração de algum já existente.

Independente se o requisito de mudança afeta estruturalmente ou semanticamente a ontologia, os tipos de conceitos que se relacionam diretamente a artefatos são os do tipo *dataTypeProperty* e do tipo *class*. Pode-se estabelecer então, a relação *dataTypeProperty-Artefato* e *class-Artefato*. Ambas serão detalhadas a seguir.

#### 4.3.1.1 Relação *dataTypeProperty*-Artefato

Nesse caso, os artefatos são relacionados diretamente ao conceito do tipo *dataTypeProperty*. Podem ser rastreadas as alterações semânticas e estruturais. Essa última ocorre quando há mudança do tipo de *range* do conceito.

A figura 43 mostra essa rastreabilidade. Nela, a *Influência da mudança (Im)* está classificada como *alta* conforme explicação constante na seção 3.6. O artefato “C-FichaBiométrica” ligado ao conceito “Meta”, teve a *Influência do conceito no artefato (Ic)* determinada como sendo *forte*. Uma vez que uma classe pertence ao diagrama de classes e, a *Influência do tipo de diagrama (Id)* foi dada como *estático*.

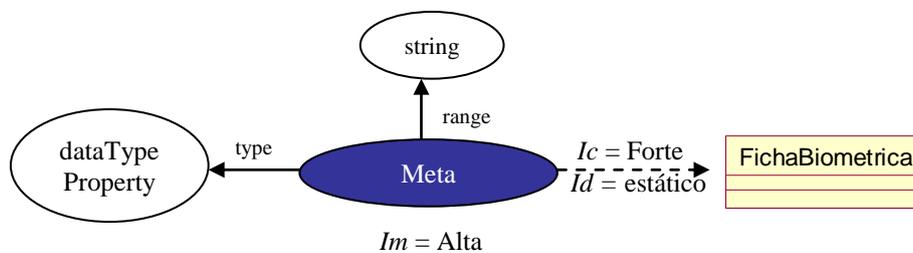


Figura 43 – Rastreabilidade de alteração de um conceito do tipo *dataTypeProperty*

Seguindo o exemplo, o *Impacto no artefato pela mudança (Iam<sub>(x,y)</sub>)* no artefato pode ser calculado como mostra a figura 44. Os valores dos elementos do conjunto representacional de *Id* são dados conforme o tipo de mudança, se de regra de negócio ou de relacionamentos na ontologia, sendo, nesse exemplo, o segundo.

$$\begin{aligned}
 Im &= \{baixa; alta\} = \{0,1; 0,9\} \\
 Ic &= \{fraco; forte\} = \{0,3; 1\} \\
 Id &= \{estático; dinâmico; funcional\} = \{1; 0,5; 0,1\} \\
 Iam_{(x,y)} &= Im \times Ic \times Id \\
 Iam_{(Meta, C-FichaBiometrica)} &= alta \times forte \times estático \\
 Iam_{(Meta, C-FichaBiometrica)} &= 0,9 \times 1 \times 1 = \mathbf{0,9}
 \end{aligned}$$

Figura 44 – Cálculos da relação *dataTypeProperty*-Artefato.

### 4.3.1.2 Relação *class*-Artefato

Os artefatos são diretamente ligados a um conceito do tipo *class*. Assim como ocorre na relação *dataTypeProperty*-Artefato, mudanças semânticas ou estruturais podem ser identificadas. Essa última ocorre quando um novo conceito do tipo *dataTypeProperty* é associado a um conceito do tipo *class*.

É necessário diferenciar os conceitos que estão sendo afetados pelo requisito de mudança, estruturalmente ou semanticamente, dos conceitos que foram identificados na ontologia do sistema, mas que não sofrem nenhuma alteração. No primeiro caso, a influência pelo tipo de mudança é *alta* e no segundo é *baixa*. Mesmo que *Influência da mudança (Im)* seja *baixa*, é interessante rastrear os artefatos relacionados a este conceito, uma vez que ele apareceu no requisito de mudança. Através da análise da variável  $Iam_{(x,y)}$ , será possível determinar a probabilidade de impacto no artefato.

A figura 45 representa a inclusão de um conceito, classificada como alteração estrutural, cuja *Influência da mudança (Im)* é tida como *alta*. Os conceitos que estão destacados são os conceitos que foram identificados no passo anterior, de identificação de conceitos. No exemplo, o conceito “dataAvaliação” está sendo associado ao conceito “FichaBiométrica”. Esse último, por sua vez, está associado a artefatos, com sua *Influência do conceito no artefato (Ic)* sendo determinada para cada um deles. É também identificado individualmente o tipo de diagrama a que o artefato pertence, utilizando a tabela VI, e estabelecendo a *Influência do tipo de diagrama (Id)*.

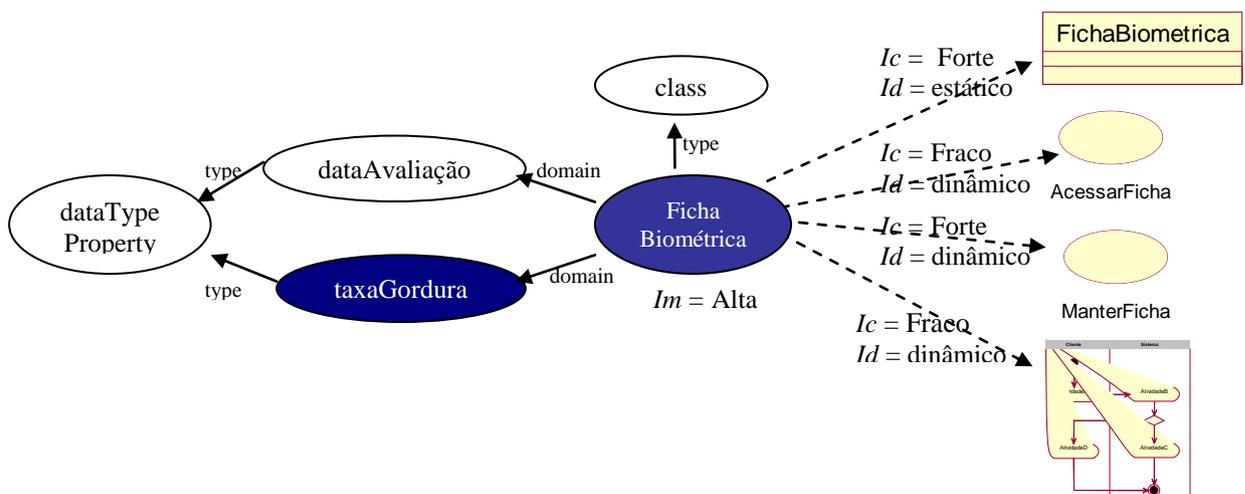


Figura 45 – Rastreabilidade de inclusão de um conceito do tipo *dataTypeProperty*

O impacto é calculado para cada artefato que for rastreado. A figura 46 mostra o cálculo da variável impacto para o artefato “UC-ManterFicha”. Os valores dos elementos dos conjuntos  $Im$  e  $Ic$  são os mesmos da figura 44.

$$\begin{aligned}
 Im &= \{baixa; alta\} = \{0,1;0,9\} \\
 Ic &= \{fraco; forte\} = \{0,3; 1\} \\
 Id &= \{estático; dinâmico; funcional\} = \{1;0,5;0,1\} \\
 \\ 
 Iam_{(x,y)} &= Im \times Ic \times Id \\
 Iam_{(FichaBiometrica, UC-ManterFicha)} &= alta \times forte \times dinâmico \\
 \mathbf{Iam}_{(FichaBiometrica, UC-ManterFicha)} &= 0,9 \times 1 \times 0,5 = \mathbf{0,45}
 \end{aligned}$$

Figura 46 – Cálculos da relação estrutural *class*-Artefato

Quando existir alteração semântica, essa deve ser indicada. A partir de então, os artefatos relacionados ao conceito podem ser analisados quanto ao impacto da mudança. A figura 47 mostra um exemplo quando o conceito “Mensalidade” está relacionado a alterações em regras de negócio.

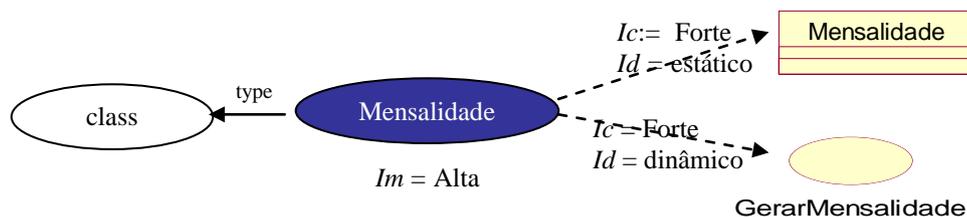


Figura 47 – Alteração de regra de negócio associada a conceito

Para realizar o cálculo desse tipo de mudança, os conjuntos de valores de  $Im$  e  $Ic$  são iguais aos apresentados na figura 44. Especificamente nesse caso, os valores dos elementos do conjunto do tipo de diagrama são diferentes. A figura 48 mostra o cálculo do artefato do tipo caso de uso “UC-GerarMensalidade”.

$$\begin{aligned}
 Im &= \{baixa; alta\} = \{0,1;0,9\} \\
 Ic &= \{fraco; forte\} = \{0,3; 1\} \\
 Id &= \{estático; dinâmico; funcional\} = \{0,5; 1; 0,1\} \\
 \\ 
 Iam_{(x,y)} &= Im \times Ic \times Id \\
 Iam_{(Mensalidade, UC-GerarMensalidade)} &= alta \times forte \times dinâmico \\
 \mathbf{Iam}_{(Mensalidade, UC-GerarMensalidade)} &= 0,9 \times 1 \times 1 = \mathbf{0,9}
 \end{aligned}$$

Figura 48 – Cálculos da relação semântica *class*-Artefato

A figura 49 destaca um conceito presente na ontologia de requisito de mudança e que foi identificado na ontologia do sistema, mas que não sofre alterações estruturais ou se-

mânticas. Apesar disso, o impacto nos artefatos pode ser avaliado, embora se considere que existe uma baixa probabilidade de serem afetados.

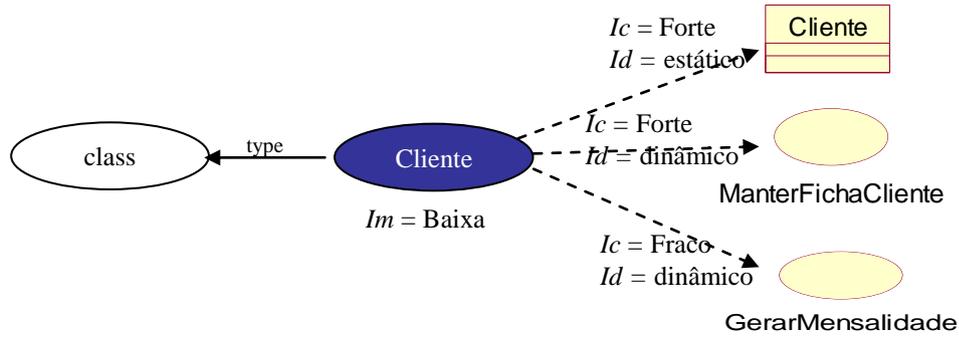


Figura 49 – Identificação de um conceito do tipo *dataTypeProperty* que não sofre mudança

O cálculo das probabilidades de impacto nos artefatos ligados no conceito “Cliente” é demonstrado na figura 50, referente ao artefato “C-Cliente”. Consideram-se os valores dos elementos do conjunto de influência do tipo de diagrama como se a alteração fosse de regra de negócio. Isso porque se infere que, se a mudança fosse estrutural, essa seria identificada ao comparar a ontologia do requisito de mudança e a ontologia do sistema. Os valores dos elementos dos conjuntos *Im*, *Ic* e *Id* são os mesmos da figura 48.

$$\begin{aligned}
 Im &= \{baixa; alta\} = \{0,1; 0,9\} \\
 Ic &= \{fraco; forte\} = \{0,3; 1\} \\
 Id &= \{estático; dinâmico; funcional\} = \{0,5; 1; 0,1\} \\
 \\ 
 Iam_{(x,y)} &= Im \times Ic \times Id \\
 Iam_{(Cliente, C-Cliente)} &= baixa \times forte \times estático \\
 \mathbf{Iam}_{(Cliente, C-Cliente)} &= 0,1 \times 1 \times 0,5 = \mathbf{0,05}
 \end{aligned}$$

Figura 50 – Cálculos da relação *class*-Artefato sem alterações.

### 4.3.2 Relação Conceito-Conceito

Associado ao conceito do tipo *objectProperty*, tem-se a *Influência do relacionamento (Ir)*, determinada por um algoritmo de *stemming*, conforme apresentado na seção 3.7.1.1. Não é necessário utilizar a influência do tipo de mudança, pois não se trata, nesse caso, de rastrear as mudanças semânticas ou estruturais a que o conceito foi submetido e sim a propagação do impacto nos conceitos. A figura 51 apresenta os artefatos que são encontrados num rastreamento de dois níveis e a relação entre os conceitos.

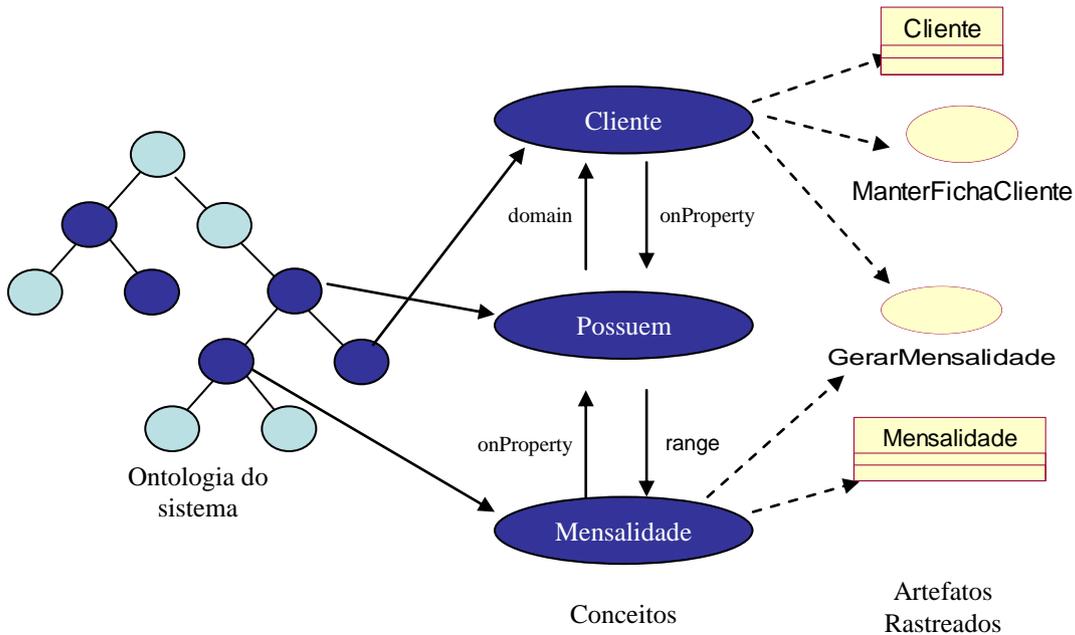


Figura 51 – Rastreabilidade de relação conceito-conceito

Considerando-se que no enésimo nível de busca na ontologia todos os artefatos do sistema serão rastreados, pois isso implica num percurso completo no grafo da ontologia, deve-se estabelecer um nível máximo de busca de conceitos nesta. A fórmula 9 calcula a *Propagação do impacto (Pi)*. A partir do resultado dessa propagação, devem ser rastreados todos os artefatos que estão relacionados a conceitos do tipo *class* e que estiverem relacionados por conceitos do tipo *objectProperty*, respeitando o critério de parada. A figura 52 apresenta uma relação conceito-conceito.

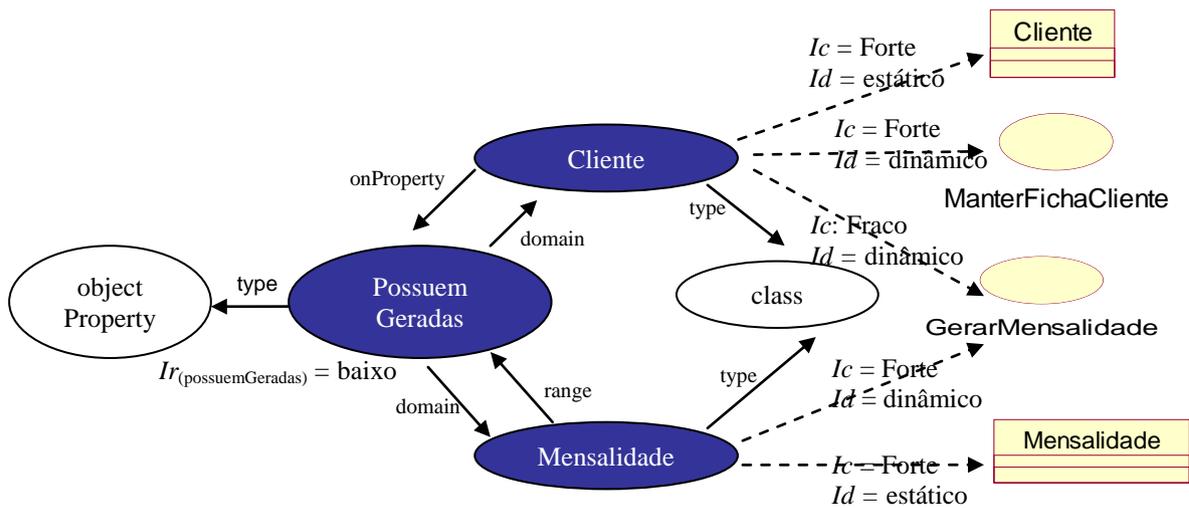


Figura 52 – Rastreabilidade de artefatos na relação conceito-conceito

O primeiro passo para analisar o impacto que se propaga entre os conceitos é determinar qual deles foi afetado com maior número de mudanças, ou seja, *Impacto Acumulado* ( $Ia_{(x)}$ ), dado pela fórmula 10. Os conceitos que serão analisados para ter essa variável calculada serão todos aqueles que foram identificados na ontologia do sistema a partir da ontologia do requisito de mudança. A figura 53 estabelece os valores do *Impacto no artefato pela mudança* ( $Iam_{(x,y)}$ ), para os conceitos e calcula o  $Ia_{(x)}$ , considerando as alterações analisadas anteriormente nas figuras 47 e 49.

$$\begin{aligned}
 Im &= \{baixa; alta\} = \{0,1; 0,9\} \\
 Ic &= \{fraco; forte\} = \{0,3; 1\} \\
 Id &= \{estático; dinâmico; funcional\} = \{0,5; 1; 0,1\} \\
 Iam_{(x,y)} &= Im \times Ic \times Id \\
 \\ 
 Iam_{(Cliente, C-Cliente)} &= 0,05 \\
 Iam_{(Cliente, UC-ManterFichaCliente)} &= 0,1 \\
 Iam_{(Cliente, UC-GerarMensalidade)} &= 0,06 \\
 \mathbf{Ia_{(Cliente)}} &= \mathbf{0,22} \\
 \\ 
 Iam_{(Mensalidade, UC-GerarMensalidade)} &= 0,45 \\
 Iam_{(Mensalidade, C-Mensalidade)} &= 0,9 \\
 \mathbf{Ia_{(Mensalidade)}} &= \mathbf{1,35}
 \end{aligned}$$

Figura 53 – Cálculo do Impacto acumulado ( $Ia_{(x)}$ )

O próximo passo é iniciar o caminhamento no grafo da ontologia a partir do conceito “Mensalidade” que teve maior *Impacto acumulado* ( $Ia_{(x)}$ ). No exemplo da figura 52, onde é mostrada parcialmente uma ontologia, o único conceito que pode ser relacionado é “Cliente”. Conforme determinado pelo algoritmo de radicalização, a *Influência do relacionamento* ( $Ir$ ) entre os dois conceitos é baixa.

Nesse exemplo, a *Propagação do impacto* ( $Pi$ ) é igual a  $Ir$ , pois só existe um conceito do tipo *objectProperty* capaz de estabelecer relacionamentos entre dois outros conceitos. Como a metodologia sugere que o critério de parada seja maior do que 70%, nesse caso não é indicado que se rastreie os artefatos do conceito “Cliente”, já que  $Pi = 0,5$ , ou seja, tem probabilidade de impacto de 50%, sendo menor do que o critério estabelecido.

#### 4.4 LISTAR ARTEFATOS

O último passo da metodologia é fornecer a lista de artefatos que tem mais probabilidade de impacto. Após ter sido calculado o impacto do artefato, seja pelo tipo de mudança ou pelo relacionamento entre conceitos, um valor é atribuído a cada um deles.

Como mostrado na figura 51, um mesmo artefato pode estar relacionado a mais de um conceito, no exemplo, o caso de uso “UC-GerarMensalidade”. Isso significa que durante o rastreamento, um mesmo artefato pode ter sido encontrado mais de uma vez e, conseqüentemente, ter o impacto calculado cada uma delas. Os impactos calculados para um mesmo artefato devem ser somados.

Após a soma dos valores de impacto dos artefatos que se repetiram, armazenada na variável *Probabilidade acumulada* ( $Pa_{(y)}$ ), onde  $y$  é o artefato, é calculado o *Percentual de probabilidade de impacto* ( $Ppi_{(y)}$ ) para cada um deles. O cálculo é dado pela fórmula 12, onde a porcentagem dos demais artefatos é normalizada a partir do maior  $Pa_{(y)}$ . Ao final, é fornecido o conjunto ordenado de forma decrescente de  $Ppi_{(y)}$  dos artefatos rastreados.

$$Ppi_{(y)} = \frac{Pa_{(y)} \times 100}{\max(Pa_{(y)})}$$

Fórmula 12 – Cálculo do Percentual de probabilidade de impacto do artefato ( $Ppi_{(y)}$ ).

Para determinar os artefatos com a maior probabilidade de impacto, é calculada a média ponderada em relação a todos aqueles que foram rastreados. Todos os artefatos que tiverem sua *Probabilidade acumulada* ( $Pa_{(y)}$ ) maior do que a média ponderada, irão formar o conjunto dos elementos que mais provavelmente serão alterados. A fórmula 13 mostra como se calcula essa média, onde  $x$  é o número de artefatos que obtiveram a mesma  $Pa_{(y)}$ .

$$\frac{Pa_{y_1} \cdot x_1 + Pa_{y_2} \cdot x_2 + Pa_{y_3} \cdot x_3 + \dots + Pa_{y_n} \cdot x_n}{Pa_{y_1} + Pa_{y_2} + Pa_{y_3} + \dots + Pa_{y_n}} = \frac{\sum_{i=1}^n (Pa_{y_i} \times x_i)}{\sum_{i=1}^n Pa_{y_i}}$$

Fórmula 13 – Cálculo da média ponderada.

Para auxiliar na aplicação da metodologia, foi proposta e desenvolvida uma ferramenta chamada ONTImpact. Essa será descrita a seguir.

## 4.5 FERRAMENTA

Com o objetivo de calcular automaticamente o impacto que os artefatos sofrem ao serem alterados por um requisito de mudança, uma ferramenta foi desenvolvida usando a tecnologia Java[JAV06]. A esse aplicativo foi dado o nome de ONTImpact, que armazena as influências dos artefatos e conceitos, compara as ontologias do requisito de mudança e do sistema e calcula o impacto nos objetos rastreados.

Para armazenar as influências que o conceito exerce sobre o artefato e o tipo de diagrama ao qual este pertence, a ontologia do sistema gerada pela ferramenta ONTrace é carregada no ONTImpact. Para que seja possível carregar os conceitos por tipos que os representam, é feita uma consulta no Jena[JEN06]. A figura 54 mostra um exemplo dessa seleção.

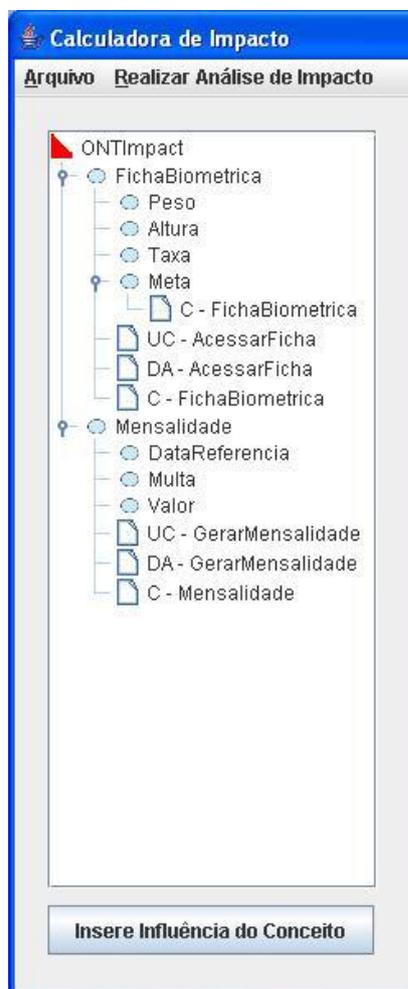


Figura 54 – Tela com os conceitos do tipo *class* e *dataTypeProperty*

Em uma janela, são apresentados os conceitos do tipo *class* e *dataTypeProperty* bem como os artefatos que foram ligados a eles. A esses objetos, é possível associar a *Influ-*

ência do conceito no artefato (*Ic*) e a *Influência do tipo de diagrama* (*Id*). A primeira pode ser classificada em: fraca, moderada e forte e é armazenada ao conceito *artifact* da ontologia, utilizando a propriedade *ontImpactConceptInfluence*. A segunda é escolhida entre estático, dinâmico e funcional, também relacionada ao conceito *artifact* e utilizando a propriedade *ontImpactDiagramInfluence*. A figura 55 mostra essa interface.

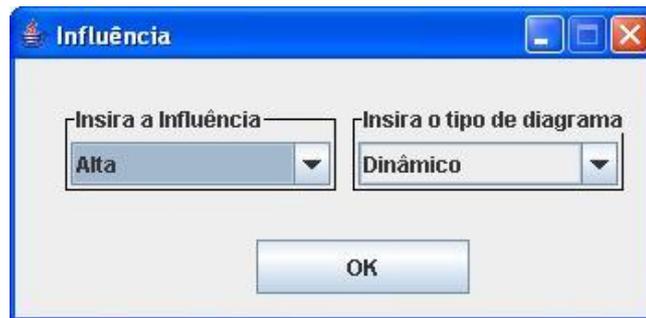


Figura 55 – Entrada das influências do conceito e tipo de diagrama

Numa segunda janela, são apresentados os conceitos do tipo *objectProperty*, juntamente com os conceitos do tipo *class* aos quais se relaciona. A esses conceitos, é atribuída a *Influência do relacionamento* (*Ir*), calculada a partir de um algoritmo de radicalização que compara os conceitos ligados a ele com as propriedades *range* e *domain*. Essa influência é então classificada em baixa ou alta e armazenada no conceito *artifact* na propriedade *ontImpactRelationshipInfluence*. A figura 56 apresenta essa tela.

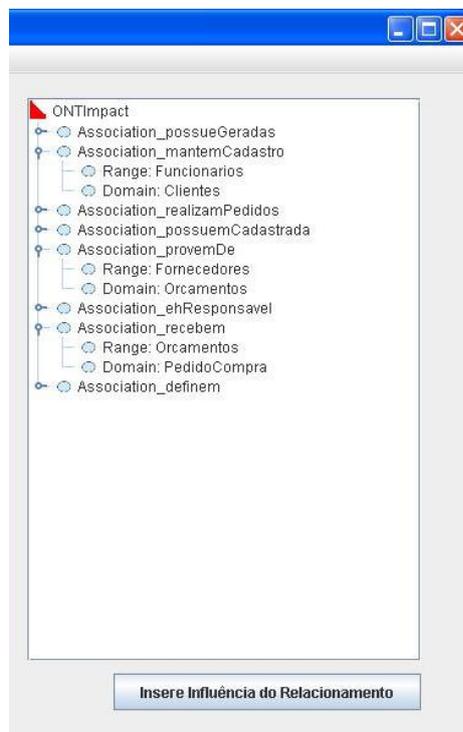


Figura 56 – Tela com os conceitos do tipo *objectProperty*

Após todas as influências terem sido atribuídas, faz-se necessário também, que o usuário do sistema informe se o conceito está tendo alguma regra de negócio associada a ele alterada. A figura 57 mostra esse procedimento. Somente os conceitos presentes na árvore de “Tipo de Mudança” podem ter essa variável setada. Essa informação é armazenada em uma variável que será utilizada no último passo da metodologia e é essencial para que a análise de impacto seja realizada com precisão.

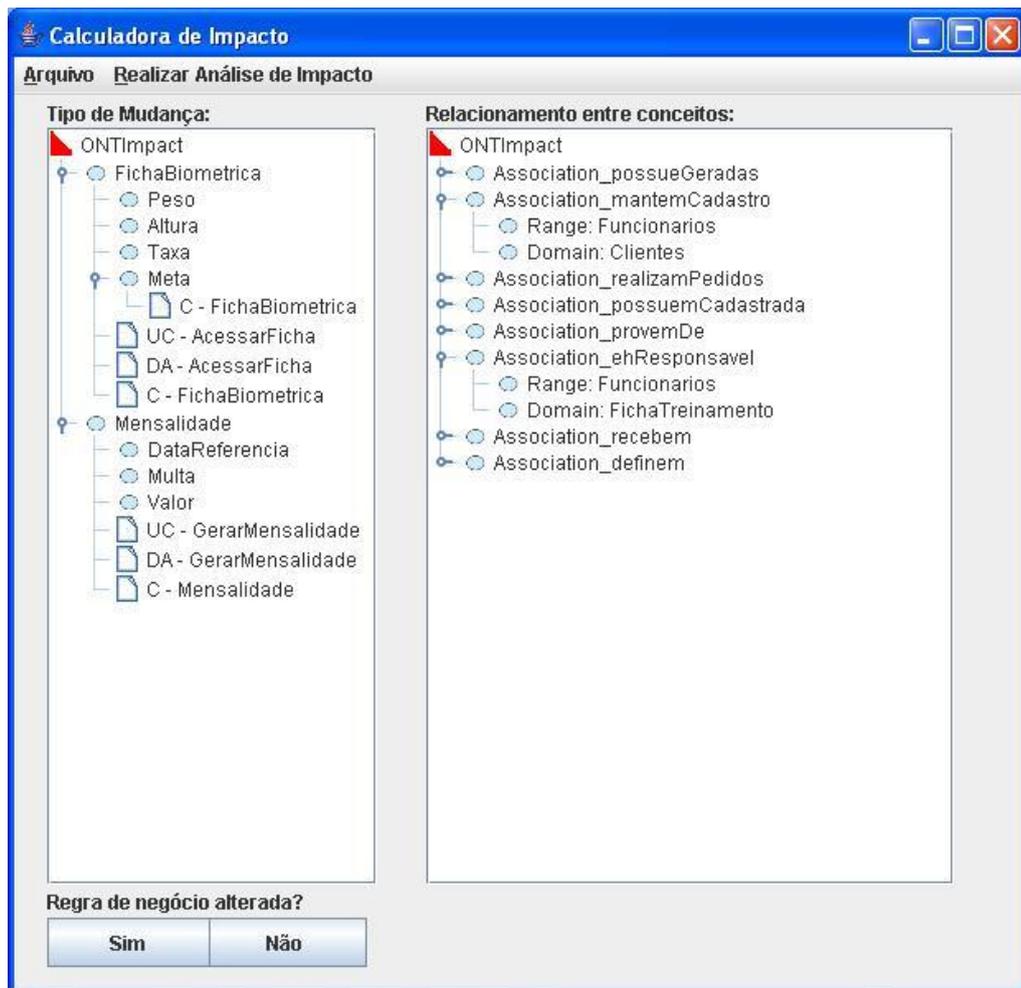


Figura 57 – Tela para informação se houve alteração de regra de negócio

Para que se faça a comparação das ontologias, o código *eXtensible Markup Language* (XML) da ontologia do requisito de mudança é carregado na ferramenta e é comparado com o código XML da ontologia do sistema. São então identificados os conceitos que são comuns as duas ontologias bem como os conceitos que estão sendo inseridos ou alterados. A figura 58 mostra a comparação entre dois códigos XML e mostra os elementos comuns identificados. No exemplo, pode-se notar que o conceito “possuemGeradas”, do tipo *objectProperty*, e os conceitos do tipo *class* “Mensalidade” e “Cliente” são comum as duas ontologias.

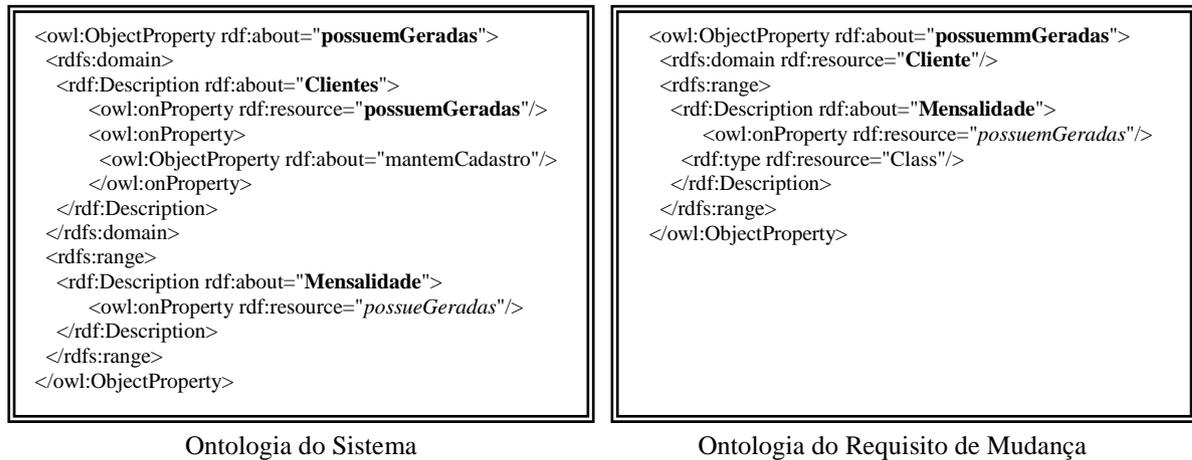


Figura 58 – Comparação de ontologias através de código XML

A última tarefa da ferramenta é realizar a análise de impacto, calculando a probabilidade de cada um dos artefatos relacionados aos conceitos serem impactados. Para isso, o usuário deve escolher entre analisar o impacto verificando as mudanças estruturais ou semânticas identificadas na ontologia ou percorrendo ela pelo relacionamento entre os conceitos. A figura 59 mostra o atalho para esse procedimento.

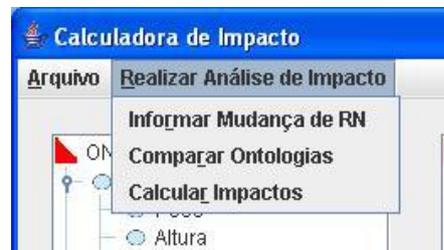


Figura 59 – Tela com menu de escolha para comparação da ontologias e cálculo do impacto

Com base nos valores dos elementos dos conjuntos de influência, são calculados o *Impacto no artefato pela mudança* ( $I_{am(x,y)}$ ) ou o *Impacto do Artefato pelo Relacionamento* ( $I_{ar(x,y)}$ ), dependendo do tipo de análise de impacto escolhida. Ao final, é apresentada uma tabela com todos os artefatos rastreados e a probabilidade de impacto para cada um deles.

#### 4.6 EXEMPLO DE APLICAÇÃO DA METODOLOGIA

O exemplo de aplicação da metodologia será apresentado utilizando o projeto *IsGym*, de controle de uma academia de ginástica, desenvolvido em uma disciplina do curso de graduação em Sistemas de Informação da Faculdade de Informática da Pontifícia Universidade Católica do Rio Grande do Sul. Esse mesmo projeto foi aplicado em um experimento

inicial dessa metodologia, que está descrito no Apêndice A, e serviu como insumo das hipóteses identificadas.

O sistema *IsGym* foi escolhido pois possui algumas premissas básicas necessárias para o desenvolvimento desse trabalho. Dentre elas está o fato de seus autores não estarem relacionados a esse estudo, ou seja, o projeto foi construído sem que seus dados fossem direcionados a resultados específicos para serem utilizados na rastreabilidade e análise de impacto. Além disso, ele foi completamente modelado seguindo o processo unificado, facilitando a abordagem aqui proposta, e teve uma implementação consistente.

#### 4.6.1 Gerar Ontologia do Requisito de Mudança

O sistema *IsGym* foi desenvolvido com a finalidade de integrar todas as funcionalidades administrativas de uma academia em um único sistema de informação. Ele mantém cadastros de clientes, funcionários, fornecedores, recursos e utilização destes, assim como as fichas de acompanhamento das avaliações (biometria) e atividades dos alunos. Foi utilizada a tecnologia *JavaServer Pages* (JSP) com conexão ao banco de dados *MS-Access* para desenvolvê-lo e a modelagem de requisitos e funcionalidades foi feita no *Rational Rose*.

Abaixo estão relacionados o dicionário de dados do sistema, o modelo conceitual construído baseado no requisito de mudança e a ontologia gerada a partir deste. O requisito de mudança, presente na figura 60, foi escolhido por incluir novos conceitos na ontologia original do sistema e alterar uma regra de negócio associada a um conceito previamente existente. Nele estão destacados os conceitos que são relevantes para analisar o problema.

Com o intuito de aumentar o número de frequentadores da academia, a *GymJoe* decidiu dar **desconto** para os **alunos** que atingirem suas **metas** de treinamento. É feita uma **avaliação** inicial do aluno. Após o resultado dessa, o aluno informa qual o percentual de **taxa de gordura** que deseja atingir nos próximos 6 meses, sendo essa a sua meta. Ao final desse **período**, uma nova avaliação é realizada e o **instrutor** verifica se o aluno atingiu seu **objetivo**. Os descontos são dados seguindo os critérios:

- 25% de desconto da **mensalidade** se o resultado atingido for maior que a meta;
- 15% se atingiu a meta ou ficou 20% abaixo desta;
- 5% se ficou entre 21% e 60% abaixo da meta

Figura 60 – Requisito de mudança para o sistema *IsGym*

Os conceitos presentes no requisito de mudança devem ser verificados no dicionário de dados do sistema para garantir a uniformidade entre as nomenclaturas. Na figura 61 estão reproduzidos os trechos importantes para esse exemplo.

<p><b>Nome:</b> Cliente;</p> <p><b>Tipo:</b> classe;</p> <p><b>Descrição:</b> descreve os dados cadastrais do cliente da academia;</p> <p><b>Pseudônimos:</b> aluno;</p> <p><b>Especificação:</b></p> <p><b>Comentários:</b> a classe está presente no modelo de domínio e no diagrama de classes.</p>
<p><b>Nome:</b> FichaBiometrica;</p> <p><b>Tipo:</b> classe;</p> <p><b>Descrição:</b> descreve os dados da avaliação do cliente;</p> <p><b>Pseudônimos:</b> avaliação;</p> <p><b>Especificação:</b></p> <p><b>Comentários:</b> a classe está presente no modelo de domínio e no diagrama de classes.</p>
<p><b>Nome:</b> Mensalidade;</p> <p><b>Tipo:</b> classe;</p> <p><b>Descrição:</b> possui atributos necessários para gerar a mensalidade;</p> <p><b>Pseudônimos:</b> fatura, boleto bancário;</p> <p><b>Especificação:</b></p> <p><b>Comentários:</b> a classe está presente no modelo de domínio e no diagrama de classes.</p>
<p><b>Nome:</b> Funcionário;</p> <p><b>Tipo:</b> classe;</p> <p><b>Descrição:</b> descreve os dados cadastrais dos funcionários;</p> <p><b>Pseudônimos:</b> professor, instrutor;</p> <p><b>Especificação:</b></p> <p><b>Comentários:</b> a classe está presente no modelo de domínio e no diagrama de classes.</p>
<p><b>Nome:</b> Meta;</p> <p><b>Tipo:</b> atributo da classe FichaBiometrica;</p> <p><b>Descrição:</b> descreve o objetivo do cliente frequentar a academia;</p> <p><b>Pseudônimos:</b> objetivo;</p> <p><b>Especificação:</b> atributo do tipo <i>String</i>;</p> <p><b>Comentários:</b> a classe FichaBiometrica está presente no modelo de domínio e no diagrama de classes.</p>
<p><b>Nome:</b> TaxaDeGordura;</p> <p><b>Tipo:</b> atributo da classe FichaBiometrica;</p> <p><b>Descrição:</b> o percentual de gordura do cliente quando avaliado;</p> <p><b>Pseudônimos:</b> índice de gordura;</p> <p><b>Especificação:</b> atributo do tipo <i>float</i></p> <p><b>Comentários:</b> a classe FichaBiometrica está presente no modelo de domínio e no diagrama de classes.</p>

Figura 61 – Dicionário de Dados *IsGym*

Após a verificação dos termos, o modelo conceitual do requisito de mudança pode ser construído. Na figura 62 está o modelo conceitual que representa o requisito de mudança.

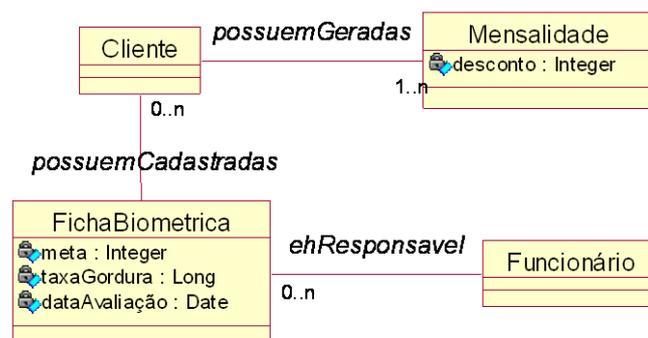


Figura 62 – Modelo de conceitual do requisito de mudança

De posse do modelo conceitual, é possível gerar a ontologia do requisito de mudança através da ferramenta ONTrace. Esta gera um código em XML, apresentado na figura 63, que é importante para identificação dos conceitos pela ferramenta que realiza a análise de impacto. A ontologia graficamente é representada pela figura 64.

```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#" >
  <owl:Ontology rdf:about="isGymAlteracao1.owl"/>
  <owl:Class rdf:about="Funcionario">
    <owl:onProperty
      <owl:ObjectProperty rdf:about="ehResponsavel"/>
    </owl:onProperty>
  </owl:Class>
  <owl:Class rdf:about="FichaBiometrica">
    <owl:onProperty
      <owl:ObjectProperty rdf:about="possuemCadastrada"/>
    </owl:onProperty>
  </owl:Class>
  <owl:Class rdf:about="Cliente">
    <owl:onProperty
      <owl:ObjectProperty rdf:about="possuemGeradas"/>
    </owl:onProperty>
  </owl:Class>
  <owl:ObjectProperty rdf:about="ehResponsavel">
    <rdfs:domain rdf:resource="Funcionario"/>
    <rdfs:range rdf:resource="FichaBiometrica"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="possuemGeradas">
    <rdfs:domain rdf:resource="Cliente"/>
    <rdfs:range>
      <rdf:Description rdf:about="Mensalidade">
        <owl:onProperty rdf:resource="possuemGeradas"/>
        <rdf:type rdf:resource="Class"/>
      </rdf:Description>
    </rdfs:range>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="possuemCadastrada">
    <rdfs:range rdf:resource="FichaBiometrica"/>
    <rdfs:domain rdf:resource="Cliente"/>
  </owl:ObjectProperty>
  <owl:DatatypeProperty rdf:about="FichaBiometrica_dataAvaliacao">
    <rdfs:domain rdf:resource="FichaBiometrica"/>
    <rdfs:range rdf:resource="date"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:about="FichaBiometrica_taxaGordura">
    <rdfs:domain rdf:resource="FichaBiometrica"/>
    <rdfs:range rdf:resource="long"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:about="FichaBiometrica_meta">
    <rdfs:domain rdf:resource="FichaBiometrica"/>
    <rdfs:range rdf:resource="int"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:about="Mensalidade_desconto">
    <rdfs:domain rdf:resource="Mensalidade"/>
    <rdfs:range rdf:resource="int"/>
  </owl:DatatypeProperty>
</rdf:RDF>

```

Figura 63 – Ontologia do requisito de mudança representada em XML.

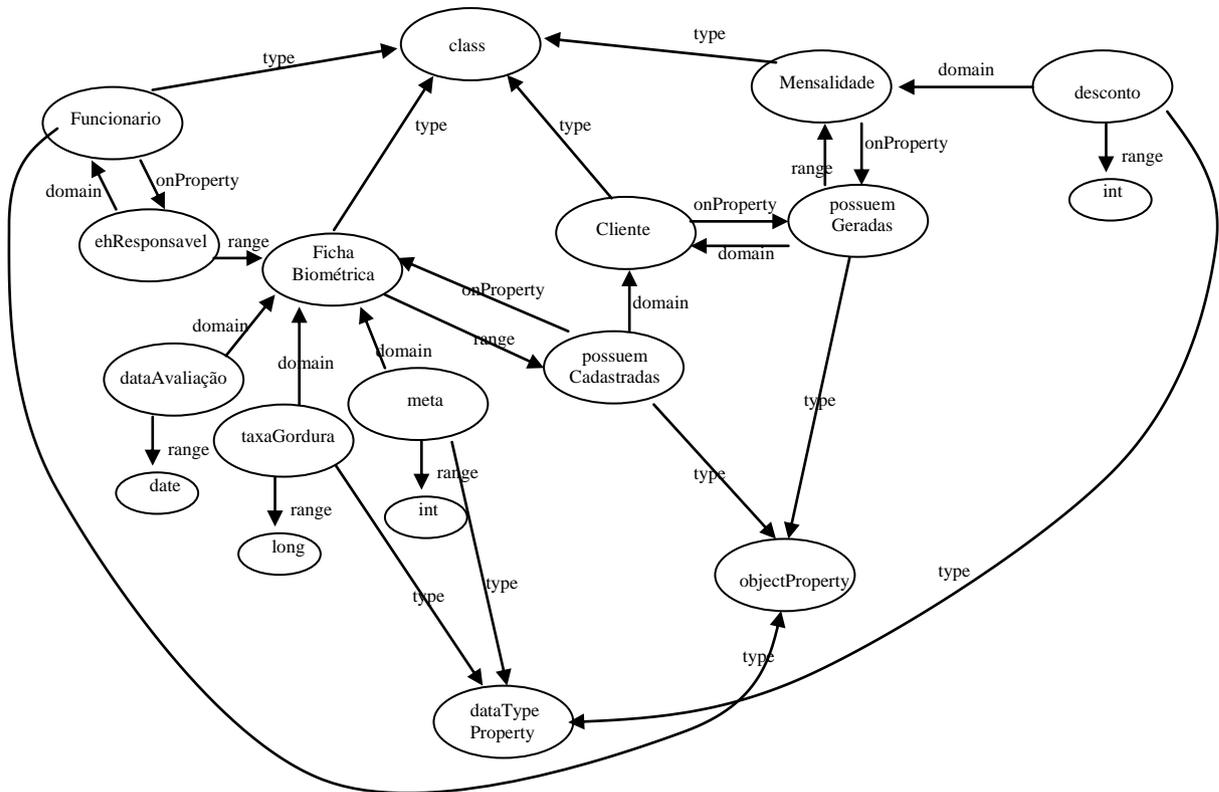


Figura 64 – Ontologia do requisito de mudança representada graficamente.

Com a ontologia gerada, pode-se passar para o próximo passo, que é de identificação dos conceitos na ontologia do sistema.

#### 4.6.2 Identificar Conceitos na Ontologia do Sistema

A comparação entre as ontologias do sistema é feita automaticamente a partir das representações XML da ontologia do requisito de mudança e da ontologia do sistema. A figura 65 representa graficamente os conceitos que são identificados na ontologia do sistema. Esta está representada parcialmente, apenas os conceitos do tipo *class* são identificados e os demais conceitos associados a este estão sendo mostrados.

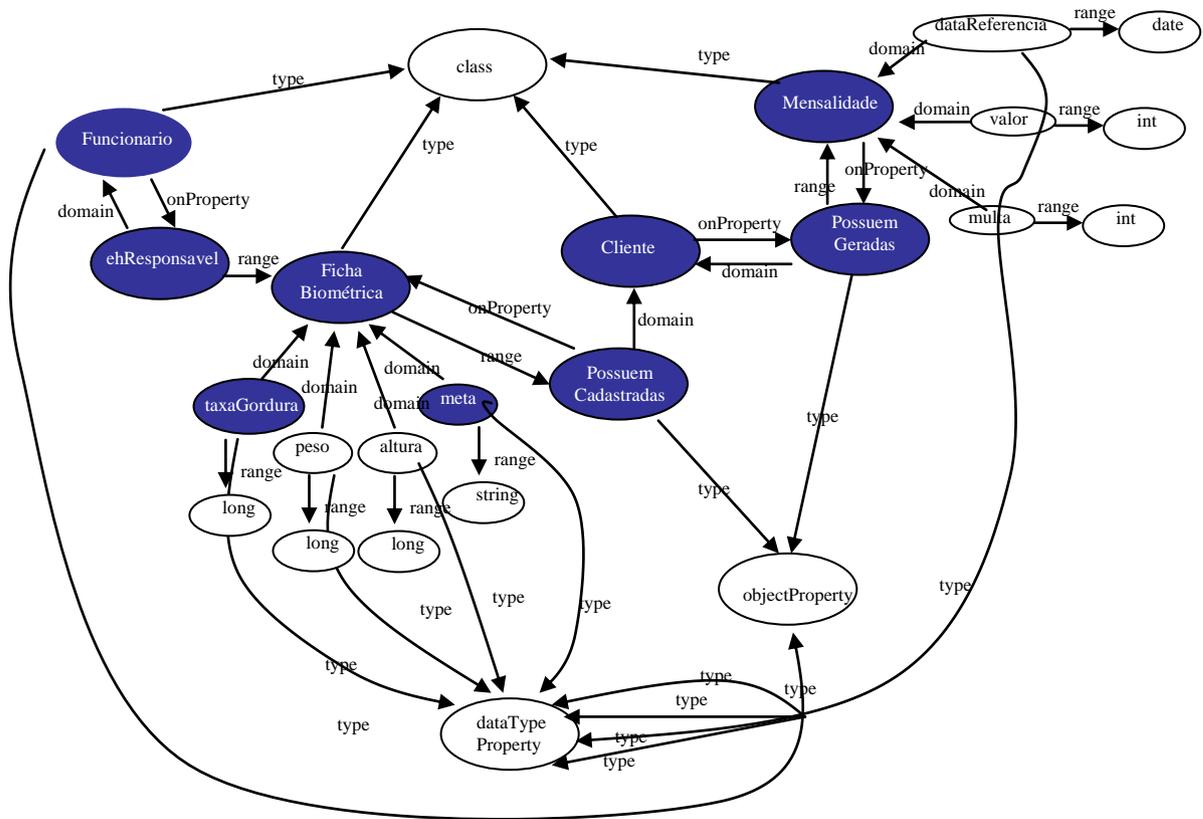


Figura 65 – Representação parcial da ontologia do sistema, com conceitos do requisito de mudança assinalados.

Foram identificadas três solicitações de alteração estrutural. Uma de alteração de um *dataTypeProperty* e duas de inclusão de *dataTypeProperty*. A primeira é a alteração do tipo do conceito “meta” ligado ao conceito “FichaBiométrica”. A figura 66 mostra essa alteração. A segunda é a inclusão do conceito “dataAvaliação” relacionado ao conceito “FichaBiométrica” e do conceito “desconto” ligado ao conceito “Mensalidade”. As figuras 67 e 68, respectivamente, apresentam essas mudanças.

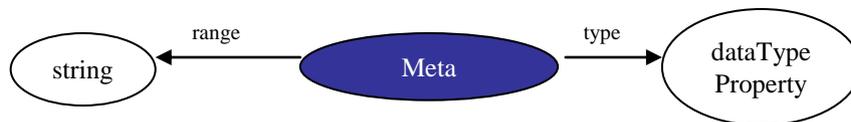


Figura 66 – Identificação do conceito “meta”.

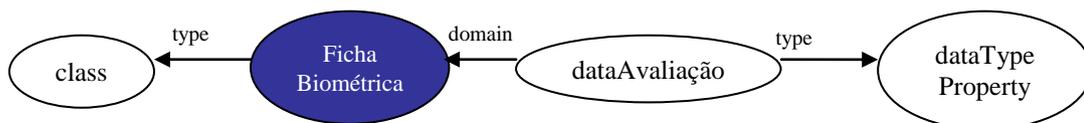


Figura 67 – Identificação do conceito “FichaBiométrica”.

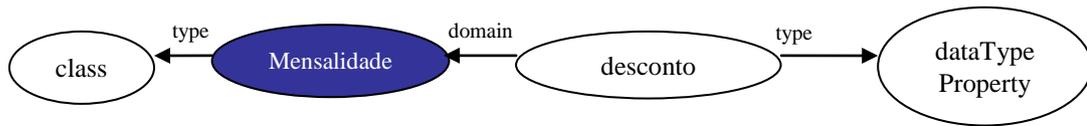


Figura 68 – Identificação do conceito “Mensalidade”, com alteração estrutural.

Houve também uma alteração de regra de negócio associada ao conceito “Mensalidade”, pois a forma de calcular a mensalidade foi alterada. Nesse caso, deve ser atribuída ao conceito a informação de que existe essa alteração, uma vez que não há como identificá-la pela comparação das ontologias. A figura 69 mostra a identificação dessa alteração.

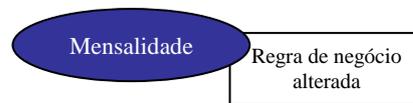


Figura 69 – Identificação do conceito “Mensalidade”, com alteração de regra de negócio.

### 4.6.3 Rastrear Artefatos

Os artefatos que foram rastreados para cada um dos conceitos identificados foram baseados na rastreabilidade feita por um especialista na ferramenta ONTrace. Primeiramente, serão rastreados e terão o impacto analisado os artefatos ligados a conceitos que sofreram alterações estruturais. Em seguida o mesmo ocorrerá para os que tiveram alteração de regra de negócio. Esses conceitos foram descritos na seção anterior. Por último, os conceitos que foram identificados, mas que não estão relacionados a nenhum desses dois tipos de mudança, serão analisados.

Será considerado, para esse projeto, que o conjunto utilizado para classificar os níveis de influência que o conceito exerce sob o artefato é de:  $\{fraco, forte\}$ , sendo os valores:  $\{0,3; 1\}$ , respectivamente. A influência do tipo de mudança é dada pelo conjunto  $\{baixa; alta\}$ , tendo valores respectivos a  $\{0,1; 0,9\}$ . A influência do tipo de diagrama é dada pelo conjunto  $\{estático; dinâmico; funcional\}$  sendo que os valores dos elementos são diferentes dependendo do tipo de mudança. Se for estrutural, é de  $\{1; 0,5; 0,1\}$  senão é de  $\{0,5; 1; 0,1\}$ .

Os artefatos rastreados para o tipo de mudança de alteração do conceito do tipo *dataTypeProperty* é dado pela figura 70. Na figura 71 é calculado o impacto do artefato, através da fórmula 8.

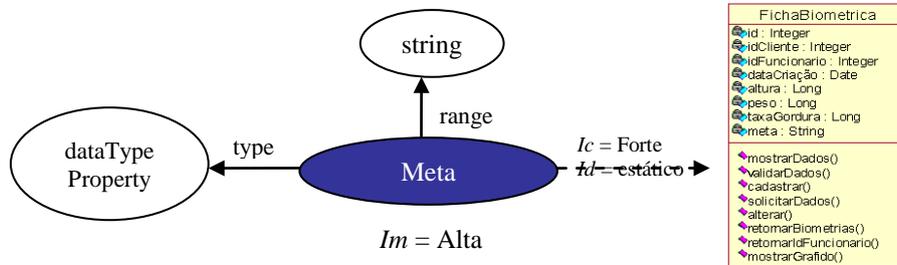


Figura 70 – Rastreamento dos artefatos ligados ao conceito “meta”.

$$Iam_{(meta,C-FichaBiometrica)} = Im \times Ic \times Id = 0,9 \times 1 \times 1 = 0,9$$

Figura 71 – Cálculo dos artefatos ligados ao conceito “meta”.

Os artefatos rastreados para o tipo de mudança de inclusão do conceito do tipo *dataTypeProperty* para o conceito “FichaBiométrica” é dado pela figura 72. Na figura 73 é calculado o impacto do artefato, através da fórmula 8.

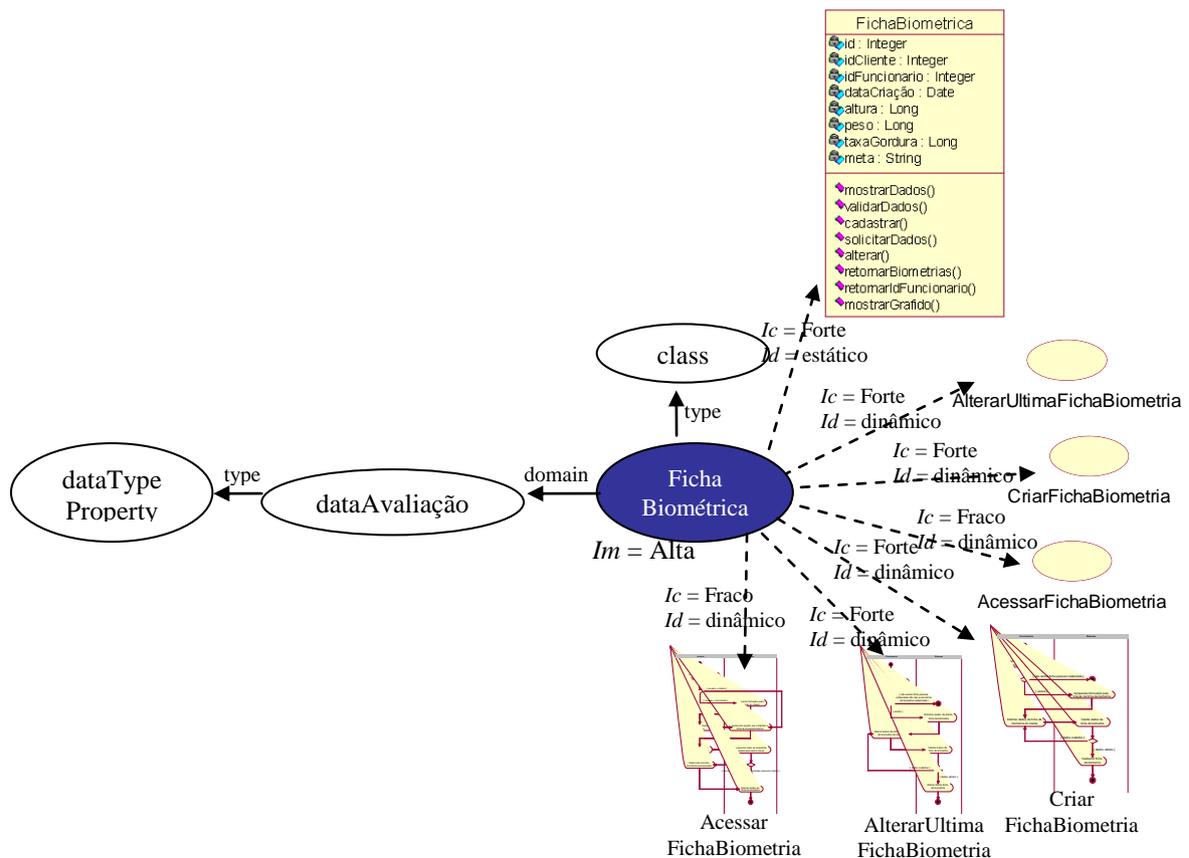


Figura 72 – Rastreamento dos artefatos ligados ao conceito “FichaBiometrica”.

$$\begin{aligned}
 Iam(FichaBiometrica,C-FichaBiometrica) &= 0,9 \times 1 \times 1 = 0,9; \\
 Iam(FichaBiometrica,UC-CriarFichaBiometrica) &= 0,9 \times 1 \times 0,5 = 0,45; \\
 Iam(FichaBiometrica,UC-AlterarUltimaFichaBiometrica) &= 0,9 \times 1 \times 0,5 = 0,45; \\
 Iam(FichaBiometrica,UC-AcessarFichaBiometrica) &= 0,9 \times 1 \times 0,5 = 0,45; \\
 Iam(FichaBiometrica,DA-CriarFichaBiometrica) &= 0,9 \times 1 \times 0,5 = 0,45; \\
 Iam(FichaBiometrica,DA-AlterarUltimaFichaBiometrica) &= 0,9 \times 1 \times 0,5 = 0,45; \\
 Iam(FichaBiometrica,DA-AcessarFichaBiometrica) &= 0,9 \times 1 \times 0,5 = 0,45;
 \end{aligned}$$

Figura 73 – Cálculos dos artefatos ligados ao conceito “FichaBiometrica”.

Os artefatos rastreado para o tipo de mudança de inclusão do conceito do tipo *dataTypeProperty* para o conceito “Mensalidade” é dado pela figura 74. Na figura 75 é calculado o impacto do artefato, através da fórmula 8.

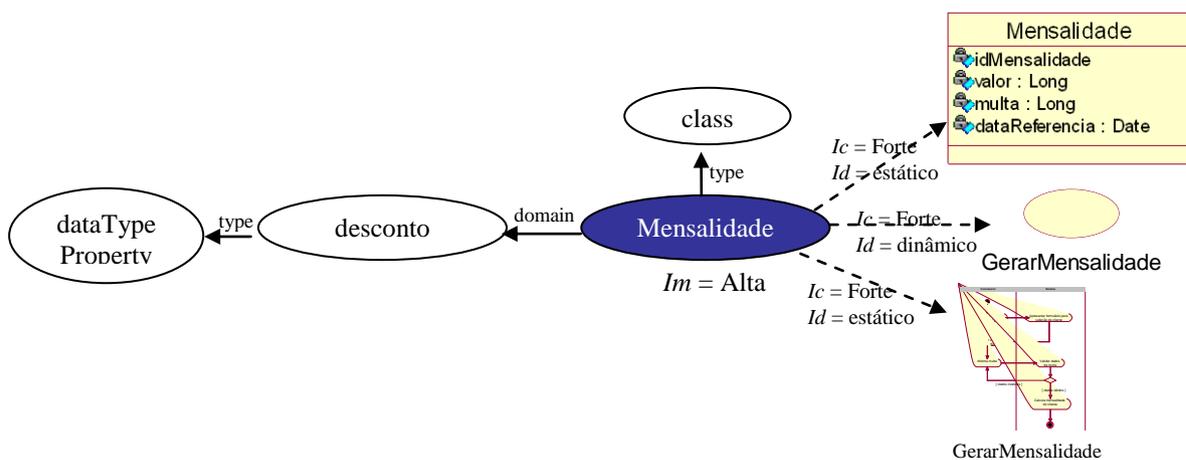


Figura 74 – Rastreamento dos artefatos ligados ao conceito “Mensalidade”.

$$\begin{aligned}
 Iam(Mensalidade,C-Mensalidade) &= 0,9 \times 1 \times 1 = 0,9; \\
 Iam(Mensalidade,UC-GenerarMensalidade) &= 0,9 \times 1 \times 0,5 = 0,45; \\
 Iam(Mensalidade,DA-GenerarMensalidade) &= 0,9 \times 1 \times 0,5 = 0,45;
 \end{aligned}$$

Figura 75 – Cálculos dos artefatos ligados ao conceito “Mensalidade” - estrutural.

Para esse mesmo conceito, existem mudanças na regra de negócio associada a ele. Os artefatos rastreados são os mesmos da figura 74, embora o cálculo do impacto seja diferente, pois os valores da variável *Id* mudam em decorrência do tipo de mudança. Na figura 76 é calculado o impacto do artefato, através da fórmula 8.

$$\begin{aligned}
 Iam(Mensalidade,C-Mensalidade) &= 0,9 \times 1 \times 0,5 = 0,45; \\
 Iam(Mensalidade,UC-GenerarMensalidade) &= 0,9 \times 1 \times 1 = 0,9; \\
 Iam(Mensalidade,DA-GenerarMensalidade) &= 0,9 \times 1 \times 1 = 0,9;
 \end{aligned}$$

Figura 76 – Cálculos dos artefatos ligados ao conceito “Mensalidade” – regra de negócio.

Os artefatos que foram identificados na ontologia do sistema a partir da ontologia do requisito de mudança, mas que não sofrem nenhum tipo de alteração, seja estrutural ou semântica, também são calculados. A finalidade é identificar a probabilidade de impacto, mesmo que pequena, dos artefatos que estão relacionados a esses conceitos. Outra motivação é que um artefato pode ser encontrado novamente e então ter seus impactos somados.

A figura 77 mostra os artefatos relacionados ao conceito “Cliente”. Na figura 78 é calculado o impacto do artefato, através da fórmula 8.

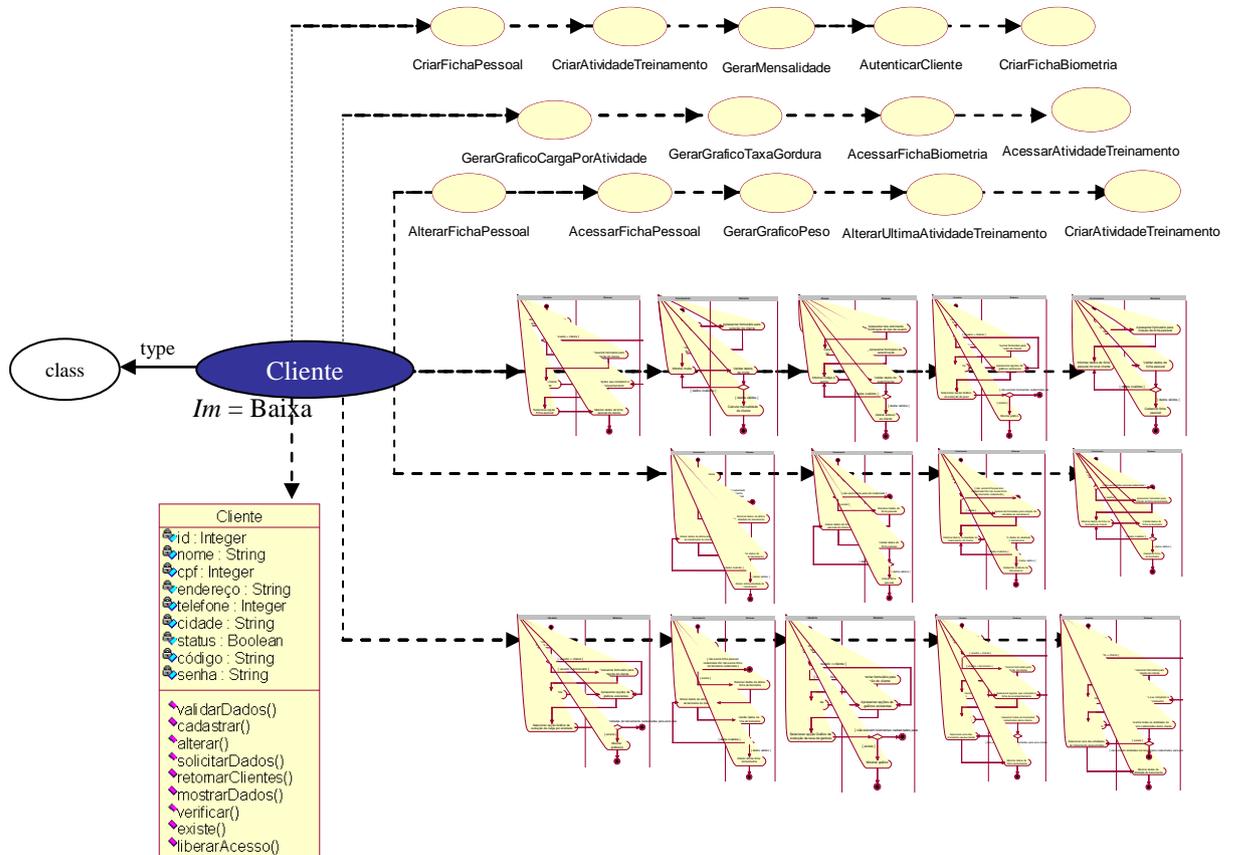


Figura 77 – Rastreamento dos artefatos ligados ao conceito “Cliente”.

$$\begin{aligned}
Iam_{(Cliente,C-Cliente)} &= 0,1 \times 1 \times 0,5 = 0,05; \\
Iam_{(Cliente,UC-CriarFichaPessoal)} &= 0,1 \times 1 \times 1 = 0,1; \\
Iam_{(Cliente,UC-AlterarFichaPessoal)} &= 0,1 \times 1 \times 1 = 0,1; \\
Iam_{(Cliente,UC-AcessarFichaPessoal)} &= 0,1 \times 1 \times 1 = 0,1; \\
Iam_{(Cliente,UC-CriarFichaBiometria)} &= 0,1 \times 1 \times 1 = 0,1; \\
Iam_{(Cliente,UC-AlterarUltimaFichaBiometria)} &= 0,1 \times 1 \times 1 = 0,1; \\
Iam_{(Cliente,UC-AcessarFichaBiometria)} &= 0,1 \times 1 \times 1 = 0,1; \\
Iam_{(Cliente,UC-CriarAtividadeTreinamento)} &= 0,1 \times 0,3 \times 1 = 0,03; \\
Iam_{(Cliente,UC-AlterarUltimaAtividadeTreinamento)} &= 0,1 \times 0,3 \times 1 = 0,03; \\
Iam_{(Cliente,UC-AcessarAtividadeTreinamento)} &= 0,1 \times 0,3 \times 1 = 0,03; \\
Iam_{(Cliente,UC-GenerarGráficoPeso)} &= 0,1 \times 0,3 \times 1 = 0,03; \\
Iam_{(Cliente,UC-GenerarGráficoTaxaGordura)} &= 0,1 \times 0,3 \times 1 = 0,03; \\
Iam_{(Cliente,UC-GenerarGráficoCargaAtividade)} &= 0,1 \times 0,3 \times 1 = 0,03; \\
Iam_{(Cliente,UC-AutenticarCliente)} &= 0,1 \times 1 \times 1 = 0,1; \\
Iam_{(Cliente,UC-GenerarMensalidade)} &= 0,1 \times 0,3 \times 1 = 0,03; \\
Iam_{(Cliente,DA-CriarFichaPessoal)} &= 0,1 \times 1 \times 1 = 0,1; \\
Iam_{(Cliente,DA-CriarFichaPessoal)} &= 0,1 \times 1 \times 1 = 0,1; \\
Iam_{(Cliente,DA-AcessarFichaPessoal)} &= 0,1 \times 1 \times 1 = 0,1; \\
Iam_{(Cliente,DA-CriarFichaBiometria)} &= 0,1 \times 1 \times 1 = 0,1; \\
Iam_{(Cliente,DA-AlterarUltimaFichaBiometria)} &= 0,1 \times 1 \times 1 = 0,1; \\
Iam_{(Cliente,DA-AcessarFichaBiometria)} &= 0,1 \times 1 \times 1 = 0,1; \\
Iam_{(Cliente,DA-CriarAtividadeTreinamento)} &= 0,1 \times 0,3 \times 1 = 0,03; \\
Iam_{(Cliente,DA-AlterarUltimaAtividadeTreinamento)} &= 0,1 \times 0,3 \times 1 = 0,03; \\
Iam_{(Cliente,DA-AcessarAtividadeTreinamento)} &= 0,1 \times 0,3 \times 1 = 0,03; \\
Iam_{(Cliente,DA-GenerarGráficoPeso)} &= 0,1 \times 0,3 \times 1 = 0,03; \\
Iam_{(Cliente,DA-GenerarGráficoTaxaGordura)} &= 0,1 \times 0,3 \times 1 = 0,03; \\
Iam_{(Cliente,DA-GenerarGráficoCarga)} &= 0,1 \times 0,3 \times 1 = 0,03; \\
Iam_{(Cliente,DA-AutenticarCliente)} &= 0,1 \times 0,3 \times 1 = 0,03; \\
Iam_{(Cliente,DA-GenerarMensalidade)} &= 0,1 \times 0,3 \times 1 = 0,03;
\end{aligned}$$

Figura 78 – Cálculos dos artefatos ligados ao conceito “Cliente”.

A figura 79 mostra os artefatos relacionados ao conceito “Funcionário”. Na figura 80 é calculado o impacto do artefato, através da fórmula 8.

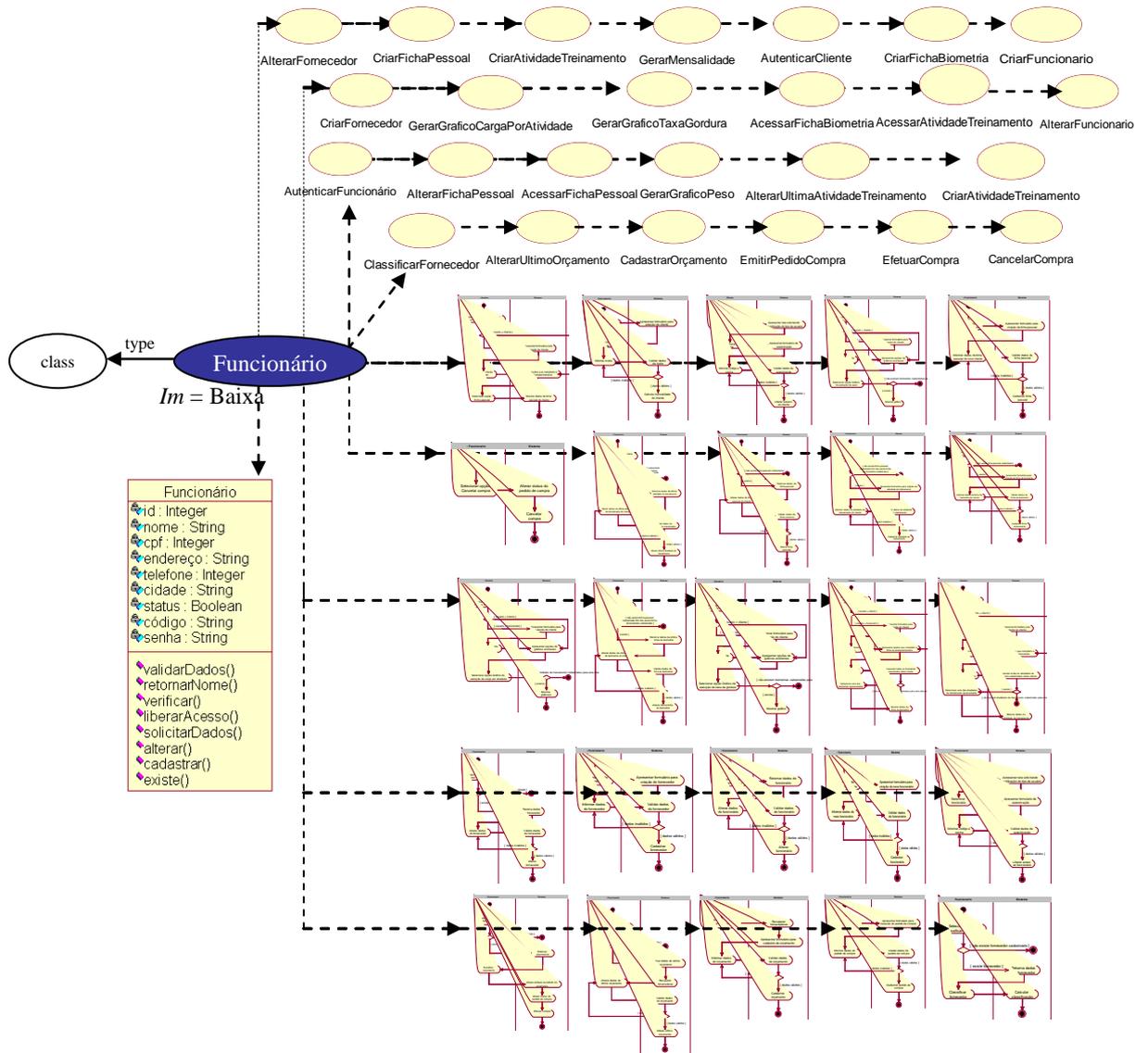


Figura 79 – Rastreamento dos artefatos ligados ao conceito “Funcionário”.

$$\begin{aligned}
&Iam_{(Funcionário,C-Funcionário)} = 0,1 \times 1 \times 0,5 = 0,05; \\
&Iam_{(Funcionário,UC-CriarFuncionário)} = 0,1 \times 1 \times 1 = 0,1; \\
&Iam_{(Funcionário,UC-AlterarFuncionário)} = 0,1 \times 1 \times 1 = 0,1; \\
&Iam_{(Funcionário,UC-CriarFornecedor)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,UC-AlterarFornecedor)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,UC-ClassificarFornecedor)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,UC-CadastrarOrçamento)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,UC-AlterarUltimoOrçamento)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,UC-CancelarCompra)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,UC-EfetuarCompra)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,UC-EmitirPedidoCompra)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,UC-CriarFichaPessoal)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,UC-AlterarFichaPessoal)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,UC-AcessarFichaPessoal)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,UC-CriarFichaBiometria)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,UC-AlterarUltimaFichaBiometria)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,UC-AcessarFichaBiometria)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,UC-CriarAtividadeTreinamento)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,UC-AlterarUltimaAtividadeTreinamento)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,UC-AcessarAtividadeTreinamento)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,UC-GenerarGráficoPeso)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,UC-GenerarGráficoTaxaGordura)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,UC-GenerarGráficoCargaAtividade)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,UC-AutenticarFuncionário)} = 0,1 \times 1 \times 1 = 0,1; \\
&Iam_{(Funcionário,UC-GenerarMensalidade)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,DA-CriarFuncionário)} = 0,1 \times 1 \times 1 = 0,1; \\
&Iam_{(Funcionário,DA-AlterarFuncionário)} = 0,1 \times 1 \times 1 = 0,1; \\
&Iam_{(Funcionário,DA-CriarFornecedor)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,DA-AlterarFornecedor)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,DA-ClassificarFornecedor)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,DA-CadastrarOrçamento)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,DA-AlterarUltimoOrçamento)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,DA-CancelarCompra)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,DA-EfetuarCompra)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,DA-EmitirPedidoCompra)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,DA-CriarFichaPessoal)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,DA-AlterarFichaPessoal)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,DA-AcessarFichaPessoal)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,DA-CriarFichaBiometria)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,DA-AlterarUltimaFichaBiometria)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,DA-AcessarFichaBiometria)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,DA-CriarAtividadeTreinamento)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,DA-AlterarUltimaAtividadeTreinamento)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,DA-AcessarAtividadeTreinamento)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,DA-GenerarGráficoPeso)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,DA-GenerarGráficoTaxaGordura)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,DA-GenerarGráficoCargaAtividade)} = 0,1 \times 0,3 \times 1 = 0,03; \\
&Iam_{(Funcionário,DA-AutenticarFuncionário)} = 0,1 \times 1 \times 1 = 0,1; \\
&Iam_{(Funcionário,DA-GenerarMensalidade)} = 0,1 \times 0,3 \times 1 = 0,03;
\end{aligned}$$

Figura 80 – Cálculos dos artefatos ligados ao conceito “Funcionário”.

#### 4.6.4 Listar e Calcular a Probabilidade dos Artefatos

O último passo é somar os valores obtidos ao calcular o impacto de cada artefato rastreado listá-los em ordem decrescente. A tabela VII mostra a probabilidade de impacto.

Tabela VII – Probabilidade de impacto dos artefatos rastreados.

Artefato	Probabilidade acumulada ( $Pa_{(y)}$ )	Percentual de probabilidade de impacto ( $Ppi_{(y)}$ ) %
C-FichaBiométrica	1,80	100,00
UC-GerarMensalidade	1,41	78,34
DA-GerarMensalidade	1,41	78,34
C-Mensalidade	1,35	75,00
UC-CriarFichaBiometria	0,58	32,22
UC-AlterarUltimaFichaBiometria	0,58	32,22
DA-CriarFichaBiometria	0,58	32,22
DA-AlterarUltimaFichaBiometria	0,58	32,22
UC-AcessarFichaBiometria	0,58	32,33
DA-AcessarFichaBiometria	0,58	32,22
UC-CriarFichaPessoal	0,13	7,22
UC-AlterarFichaPessoal	0,13	7,22
UC-AcessarFichaPessoal	0,13	7,22
UC-AuthenticarCliente	0,13	7,22
DA-CriarFichaPessoal	0,13	7,22
DA-AlterarFichaPessoal	0,13	7,22
DA-AcessarFichaPessoal	0,13	7,22
DA-AuthenticarCliente	0,10	5,56
UC-CriarFuncionário	0,10	5,56
UC-AlterarFuncionário	0,10	5,56
UC-AuthenticarFuncionário	0,10	5,56
DA-AuthenticarFuncionário	0,10	5,56
UC-CriarAtividadeTreinamento	0,06	3,33
UC-AlterarUltimaAtividadeTreinamento	0,06	3,33
UC-AcessarAtividadeTreinamento	0,06	3,33
UC-GerarGráficoPeso	0,06	3,33
UC-GerarGráficoTaxaGordura	0,06	3,33
UC-GerarGráficoCargaAtividade	0,06	3,33
DA-CriarAtividadeTreinamento	0,06	3,33
DA-AlterarUltimaAtividadeTreinamento	0,06	3,33
DA-AcessarAtividadeTreinamento	0,06	3,33
C-Cliente	0,05	2,78
C-Funcionário	0,05	2,78
UC-CriarFornecedor	0,03	1,67
UC-AlterarFornecedor	0,03	1,67
UC-ClassificarFornecedor	0,03	1,67
UC-CadastrarOrçamento	0,03	1,67
UC-AlterarUltimoOrçamento	0,03	1,67
UC-CancelarCompra	0,03	1,67
UC-EfetuarCompra	0,03	1,67
UC-EmitirPedidoCompra	0,03	1,67
DA-CriarFuncionário	0,03	1,67
DA-AlterarFuncionário	0,03	1,67
DA-CriarFornecedor	0,03	1,67
DA-AlterarFornecedor	0,03	1,67
DA-ClassificarFornecedor	0,03	1,67
DA-CadastrarOrçamento	0,03	1,67
DA-AlterarUltimoOrçamento	0,03	1,67
DA-CancelarCompra	0,03	1,67
DA-EfetuarCompra	0,03	1,67
DA-EmitirPedidoCompra	0,03	1,67

Na tabela VII foram destacados os artefatos que tem maior probabilidade de impacto. Esse conjunto de artefatos com maior probabilidade de impacto é formado por aqueles que tiverem a *Probabilidade acumulada* ( $Pa_{(y)}$ ) maior do que a média ponderada calculada para todos os elementos rastreados. A figura 81 apresenta sua aplicação nesse exemplo.

$$\frac{1 \times 1,8 + 2 \times 1,41 + 1 \times 1,35 + 6 \times 0,58 + 7 \times 0,13 + 5 \times 0,1 + 9 \times 0,06 + 2 \times 0,05 + 18 \times 0,03}{1 + 2 + 1 + 6 + 7 + 5 + 9 + 2 + 18} = 0,23$$

Figura 81 – Cálculo da média ponderada.

Nota-se que a classe “FichaBiométrica”, que atingiu o maior impacto acumulado foi considerada como o artefato que com certeza seria impactado, isto é, probabilidade de 100%. A probabilidade dos demais artefatos foi calculada dentro dessa escala. Os resultados aqui apresentados podem ser verificados no experimento constante no Apêndice A, onde se comprova que os objetos mais atingidos estão entre os identificados pela metodologia de análise de impacto baseada em ontologia.

Para o sistema *IsGym*, não é possível aplicar a influência do relacionamento entre conceitos, pois com o uso do algoritmo de radicalização, todos os relacionamentos entre conceitos são baixos. Nesse quesito, o sistema deveria ter sido desenvolvido com maior preocupação na denominação dos conceitos.



## 5 VALIDAÇÃO DA METODOLOGIA

Esse capítulo apresenta a validação da proposta de metodologia de análise de impacto baseada na rastreabilidade ontológica. Para realizá-la, foi utilizada a Engenharia de Software Experimental.

Igualmente ao primeiro experimento realizado, que auxiliou na definição dessa proposta, o sistema de software utilizado no experimento de validação da metodologia é o *IsGym*. Este é amplamente apresentado no Apêndice A. Foram utilizados também os mesmos requisitos de mudança usados no primeiro experimento, mostrados nas figuras 98 e 99. Isso foi possível, pois os participantes de um e outro experimento são diferentes.

É importante destacar que esse experimento, no que diz respeito ao sistema de software utilizado, bem como alguns pontos do planejamento e seleção de participantes foi feito em conjunto ao experimento de validação da proposta de rastreabilidade ontológica [NOL05b]. Sendo assim, algumas características de ambos os experimentos serão as mesmas ou semelhantes. A seguir, serão apresentados os passos necessários para se construir e validar um experimento, como consta no referencial teórico desse trabalho.

### 5.1 DEFINIÇÃO

Para estabelecer os objetivos desse estudo, foi utilizada a técnica GQM proposta por Basili [BAS94]. Nessa etapa, é necessário apresentar as questões que serão avaliadas e as métricas que serão utilizadas para medi-las.

### 5.1.1 Objetivo Global

Avaliar a precisão da análise de impacto baseada na rastreabilidade indexada por conceitos da ontologia com relação à proposta de análise de impacto baseada na rastreabilidade indexada por requisitos [ONE03].

### 5.1.2 Objetivo do Estudo

*Comparar a metodologia de análise de impacto baseada em rastreabilidade indexada por conceitos e por requisitos,*

*Com o propósito de caracterizar, com base em um requisito de mudança, a de probabilidade de alteração de artefatos e o conjunto de artefatos realmente modificados,*

*Com foco na precisão,*

*Sob o ponto de vista do analista de software,*

*No contexto de manutenção de um sistema de informação no domínio de academia desportiva, desenvolvido por estudantes durante uma disciplina da Faculdade de Informática da PUCRS.*

### 5.1.3 Objetivo da Medição

Qual a precisão, indicada pela intersecção entre o conjunto de artefatos previstos para alteração e os realmente alterados, das abordagens de análise de impacto baseada em ontologias e baseada em requisitos.

### 5.1.4 Questão

A precisão do conjunto dos artefatos com maior probabilidade de impacto encontrada através da análise de impacto baseada na rastreabilidade através de requisitos é igual a precisão do conjunto dos artefatos com maior probabilidade de impacto encontrada através da análise de impacto baseada na rastreabilidade ontológica?

### 5.1.5 Métricas

A métrica utilizada para responder a questão estabelecida corresponde à precisão de cada uma das abordagens, com relação à completude e corretude das previsões de alteração de artefatos em relação aos artefatos que foram realmente modificados. Por precisão, nesse experimento, considera-se a razão entre o conjunto de artefatos previstos e modificados e o conjunto total de artefatos previstos e modificados. O conjunto de artefatos previstos e modificados é composto pelo conjunto dos artefatos com maior probabilidade de impacto, estabelecidos através da aplicação de cada uma das metodologias de análise de impacto, interseccionados com o conjunto dos artefatos realmente modificados. O conjunto total de artefatos previstos ou modificados é composto por todos os artefatos que tiveram previsão de impacto, e por todos os artefatos que foram realmente modificados. O cálculo da precisão está representado na fórmula 14.

$$P = \frac{qtdArtefatos(Pb \cap M)}{qtdArtefatos(Pb \cup M)}$$

Fórmula 14 – Cálculo da variável Precisão.

Onde:

*M*: Conjunto de artefatos modificados;

*Pb*: Conjunto de artefatos previstos por determinada metodologia de análise de impacto;

*qtdArtefatos()*: função que recupera a quantidade de artefatos do conjunto.

## 5.2 PLANEJAMENTO

O projeto do experimento é determinado, a instrumentação é considerada e os aspectos da validade do experimento são avaliados. Realiza-se a seleção do contexto, formulam-se as hipóteses, selecionam-se as variáveis e os participantes, projeta-se o experimento, realiza-se a preparação conceitual da instrumentação e considera-se a validade do experimento.

### 5.2.1 Seleção do Contexto

Para a condução do experimento de análise de impacto, foi escolhido o contexto de uma universidade e não de um ambiente realista, pois esse é mais arriscado e envolve custos não previstos nesse experimento. Também fazem parte do contexto:

- a) Processo: será utilizada à abordagem *In-vitro*, na qual o conjunto de participantes executará o experimento em um ambiente controlado. Este experimento não se dará durante o desenvolvimento de software industrial, isto é, ele será *off-line*;
- b) Participantes: o experimento será conduzido por alunos de graduação e pós-graduação da Faculdade de Informática da PUCRS;
- c) Realidade: o problema estudado será de sala de aula (*toy example*) e corresponde a um sistema modelado e desenvolvido por alunos da graduação, durante uma disciplina da Faculdade de Informática da PUCRS, no domínio de uma academia desportiva. O objetivo desta escolha é a utilização de um sistema que se aproxime do real e que foi desenvolvido sem a intervenção do pesquisador;
- d) Generalidade: o experimento é específico e com validade apenas no escopo do presente estudo.

## 5.2.2 Formulação das Hipóteses

Para o experimento, foi definida informalmente a hipótese:

*Sugere-se que a análise de impacto baseada na rastreabilidade por requisitos prevê o mesmo conjunto de artefatos que a análise de impacto baseada por conceitos de uma ontologia, com relação ao conjunto realmente modificado.*

A formalização das hipóteses e a definição de suas medidas se dão por:

**Hipótese Nula,  $H_0$ :** A precisão na previsão de artefatos impactados por uma mudança obtidos pela análise de impacto baseada na rastreabilidade por requisitos é igual a análise de impacto baseada na rastreabilidade por conceitos da ontologia.

a. **Medidas:** A precisão será avaliada pela relação entre o conjunto de artefatos modificados e que tinham sido previstos e o conjunto de união dos artefatos que foram previstos ou modificados, onde:

i.  $P_{air}$ : Precisão associada à análise de impacto baseada na rastreabilidade por requisitos;

ii.  $P_{aic}$ : Precisão associada à análise de impacto baseada na rastreabilidade por conceitos da ontologia;

b.  **$H_0$ :**  $P_{air} = P_{aic}$

c. **Hipótese Alternativa,  $H_1$ :** A precisão na previsão dos artefatos obtidos pela análise de impacto baseada na rastreabilidade por requisitos é maior do que a da análise de impacto baseada na rastreabilidade por conceitos da ontologia.

$$H_1: P_{air} > P_{aic}$$

d. **Hipótese Alternativa,  $H_2$ :** A precisão na previsão dos artefatos obtidos pela análise de impacto baseada na rastreabilidade por conceitos da ontologia é maior do que a da análise de impacto baseada na rastreabilidade por requisitos.

$$H_2: P_{aic} > P_{air}$$

### 5.2.3 Seleção das Variáveis

As variáveis independentes servem como entrada para o experimento, enquanto as dependentes se referem à saída.

#### 5.2.3.1 Variável Independente

Assumiram-se como variáveis independentes:

- a) Experiência do time de manutenção;
- b) Metodologia de análise de impacto.

#### 5.2.3.2 Variável Dependente

Assume-se como variável dependente:

- a) *Precisão* através da relação entre os conjuntos de artefatos previstos e modificados e do total de artefatos previstos ou modificados.

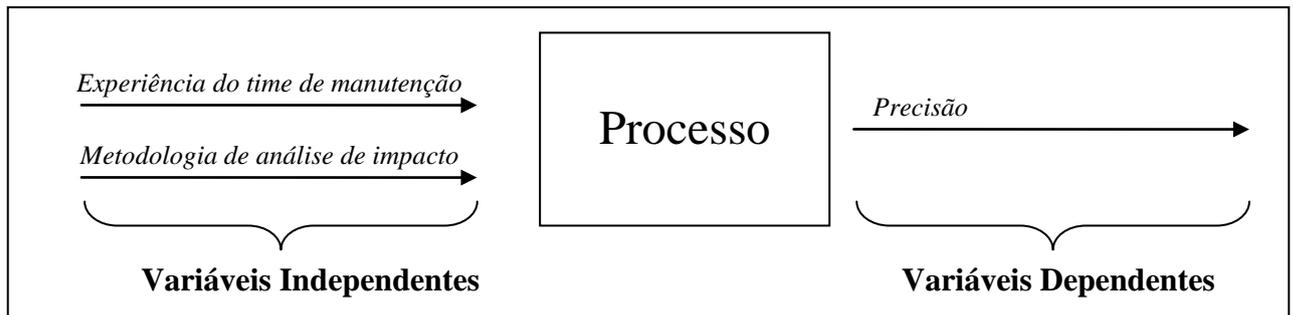


Figura 82 – Variáveis independentes e dependentes do estudo experimental.

#### **5.2.4 Seleção dos Indivíduos**

A população definida para o experimento é formada por alunos do curso de graduação e pós-graduação da Faculdade de Informática da PUCRS, totalizando doze alunos.

Não será utilizada uma amostragem probabilística para seleção dos indivíduos, mas sim uma amostragem não probabilística:

- a) Amostragem por conveniência: serão escolhidas as pessoas mais convenientes para o experimento.

#### **5.2.5 Projeto do Experimento**

Caracteriza a forma de condução do experimento, decidindo, por exemplo, a alocação dos participantes.

##### **5.2.5.1 Princípios Genéricos de Projetos**

Dentre os princípios genéricos para o projeto do experimento, caracteriza-se:

- a) Aleatoriedade: a aleatoriedade será utilizada para definir quais participantes irão executar cada metodologia de análise de impacto (baseada na rastreabilidade através de requisito ou conceito);
- b) Obstrução: durante a experimentação, muitos dos participantes não possuem o mesmo nível de experiência acadêmica e profissional. Para minimizar o efeito da experiência sobre o experimento, os indivíduos foram selecionados utilizando o critério de quota e conveniência;
- c) Balanceamento: este princípio será utilizado no experimento para que cada proposta de análise de impacto seja executada pela mesma quantidade de participantes.

### 5.2.5.2 Padrão para Tipo de Projeto

Para a hipótese que será validada, serão utilizadas as seguintes notações:

$\mu_{air}$  Análise de impacto baseada na rastreabilidade por requisitos;

$\mu_{aic}$  Análise de impacto baseada na rastreabilidade por conceitos.

Esse experimento investiga se a precisão de  $\mu_{air}$  é igual a precisão de  $\mu_{aic}$ . A abordagem conhecida como *um fator com dois tratamentos*. Nesse caso, o fator refere-se ao time de manutenção, ou seja, aos participantes do experimento e os tratamentos são as metodologias de análise de impacto que estão sendo comparadas, a baseada em rastreabilidade por requisitos e por conceitos.

Optou-se por conduzir o projeto do experimento, no que diz respeito a execução desse, utilizando uma abordagem completamente aleatória para escolha de quais participantes iriam executar uma ou outra metodologia. Essa decisão foi tomada, pois se cada participante fizesse uso das duas metodologias, os resultados quando a segunda metodologia fosse utilizada seriam prejudicados, uma vez que a pessoa já teria um conhecimento prévio do sistema e do requisito de mudança que estaria sendo proposto. A tabela VIII, chamada de tabela de contingência, mostra com os fatores (participantes) foram divididos para executar os dois tratamentos (metodologias).

Tabela VIII – Tabela de contingência.

#Participante	$\mu_{aic}$	$\mu_{air}$
1	X	
2		X
3	X	
4	X	
5		X
6		X
7	X	
8		X
9	X	
10		X
11	X	
12		X
11	X	
12		X

Para verificar a hipótese, alguns testes de significância são sugeridos na literatura. O Teste T é usado para realizar um teste paramétrico com duas amostras independentes. Já o

teste *Mann-Whitney* é usado caso o teste seja não-paramétrico. Dependendo da normalidade e variância dos dados obtidos na execução, será escolhido um ou outro teste.

### 5.2.6 Instrumentação

Para realizar o experimento, será utilizado:

- a) **Objetos:** a descrição do sistema de informação *IsGym*, sua modelagem UML do diagrama de classes de projeto, a descrição dos casos de uso e o Modelo de Domínio, além de dois requisitos de mudança. Para auxiliar na análise de impacto, será fornecida para cada abordagem uma planilha onde deverão se determinadas as variáveis usadas para calcular o conjunto de artefatos com maior probabilidade de impacto para cada metodologia, e serem informados os artefatos que foram modificados;
- b) **Guias:** será fornecido um treinamento para os participantes, apresentando as duas metodologias de análise de impacto, o contexto do experimento e a motivação da equipe. Também será fornecido um tutorial sobre como proceder durante o experimento;
- c) **Métricas:** Os dados serão recuperados através da planilha preenchida pelos participantes e das alterações feitas diretamente no diagrama de classes do sistema, utilizando para isso a ferramenta *ArgoUML*.

### 5.2.7 Análise da Validade

Os resultados devem ser válidos para a população a qual os participantes fazem parte. São definidas as validades interna, externa, construção e conclusão.

### 5.2.7.1 Validade Interna

Serão avaliados alguns critérios, tais como:

- a) *Histórico*: a data de aplicação do experimento será criteriosamente definida, evitando períodos nos quais os participantes possam sofrer influências externas;
- b) *Maturação*: durante o treinamento, serão utilizadas técnicas de motivação para incentivar positivamente os participantes;
- c) *Seleção dos grupos*: será utilizada uma abordagem para nivelar o conhecimento dos participantes através de um treinamento sobre as metodologias. A execução das atividades será individual;
- d) *Difusão*: durante o treinamento, será desenvolvida uma motivação que não incentive interação entre os participantes. Adicionalmente, haverá um policiamento durante a experimentação para evitar este tipo de interação.

### 5.2.7.2 Validade Externa

Para esta avaliação, será adotada a interação da seleção, ou seja, os participantes que foram selecionados possuem um perfil apto aos tratamentos do experimento, apresentando, em sua maioria, conhecimento prévio sobre processo de desenvolvimento de software e modelagem de sistemas, além de experiência em indústria.

### 5.2.7.3 Validade de Construção

Durante o experimento, serão avaliados:

- a) *Inadequada explicação pré-operacional*: consiste na explicação operacional do experimento, visando mostrar como a metodologia é aplicada;
- b) *Adivinhação de hipóteses*: devido ao fato dos participantes serem humanos, é possível sua interação com o experimento, sugerindo novas hipóteses e exercitando a criatividade. É importante manter o foco no estudo planejado;

- c) *Expectativas do condutor do experimento*: ao se conduzir um experimento, o responsável pode exercer influências sobre as variáveis envolvidas e sobre o material elaborado. Durante a presente proposta, todo o material utilizado será previamente avaliado por outro responsável.

#### 5.2.7.4 Validade de Conclusão

A validade de conclusão segue as perspectivas:

- a) *Manipulação dos dados*: como os dados resultantes do experimento serão manipulados pelo pesquisador, é possível que os mesmos sofram algumas variações, tal como o coeficiente de significância para validação dos resultados;
- b) *Confiabilidade das medidas*: sugere que medidas subjetivas possam ser influenciadas pelo pesquisador. Essa perspectiva será minimizada já que as medidas foram definidas sem dependência do critério humano;
- c) *Confiabilidade na implementação dos tratamentos*: consiste no risco em que diferentes participantes possam implementar de forma distinta os processos estabelecidos pelo experimento. Este risco não será evitado, uma vez que a determinação das variáveis das metodologias de análise de impacto comparadas é subjetiva e dependente da interpretação de cada participante no experimento, assim como a modelagem de cada requisito de mudança propostos. Com isso, os participantes poderão prever e modificar artefatos diferentes;
- d) *Configurações do ambiente do experimento*: consiste nas interferências externas do ambiente que podem influenciar os resultados durante a execução do experimento. O experimento será executado em um laboratório isolado, onde será proibida a interação externa como celulares, saídas, etc.;
- e) *Heterogeneidade aleatória dos participantes*: a escolha de diferentes participantes com diferentes experiências pode exercer um risco na variação dos resultados.

## 5.3 EXECUÇÃO

Os dados são coletados e é feita a validação preliminar destes.

### 5.3.1 Preparação

Foram analisadas algumas características intrínsecas à execução:

- a) *Consenso com o experimento*: se os participantes não concordam com os objetivos da pesquisa ou não tem conhecimento sobre o experimento, corre-se o risco de que sua participação não ocorra em encontro aos objetivos. Durante a experimentação, a preparação dos participantes deverá fornecer o embasamento necessário sobre o experimento, clarificando quais os objetivos e metas almejadas;
- b) *Resultados sensitivos*: é possível que o resultado obtido pelo experimento se influencie por questões pessoais, como a sensibilidade dos participantes por estarem sendo avaliados. Será adotada uma postura de anonimato dos participantes em toda a descrição da experimentação.

O primeiro passo para execução do experimento foi a apresentação de um treinamento específico para cada grupo, demonstrando cada uma das metodologias de análise de impacto. Nesse mesmo momento, foram explanados os objetivos, a técnica, a motivação e o procedimento técnico para condução do experimento. Após, os participantes receberam a instrumentação prevista, fazendo parte dessa um tutorial contendo o passo-a-passo para a execução das atividades. Os participantes do experimento foram responsáveis pela coleta dos dados, preenchendo a planilha que foi disponibilizada. Todo esse material está presente no Apêndice B.

### 5.3.2 Execução

A execução do experimento para cada metodologia foi feita em turnos diferentes, sem tempo determinado para o término das atividades. O pesquisador esteve presente durante toda a duração do experimento a fim de conduzi-lo, ficando a disposição para esclarecimento de eventuais dúvidas, embora sem interferência nos dados que estavam sendo coletados.

Em ambas as metodologias, a base é a rastreabilidade dos artefatos, seja através de casos de uso ou de conceitos da ontologia. Dessa forma, para cada uma delas foi fornecido o gabarito da rastreabilidade, ou seja, para a metodologia de análise de impacto baseada na rastreabilidade por requisitos, foram disponibilizadas todas as classes que se associam a um caso de uso. Para a metodologia de análise de impacto baseada na rastreabilidade por conceitos, foram fornecidas as classes indexadas por conceitos. A partir dessas informações e, após o entendimento do requisito de mudança proposto, os participantes foram capazes de identificar a qual caso de uso ou conceito o requisito de mudança se referia. Após foi possível determinar os valores das variáveis necessárias para cada metodologia, obtendo assim o conjunto de artefatos com maior probabilidade de impacto. O último passo era realizar efetivamente a manutenção no diagrama de classes.

## 5.4 ANÁLISE E INTERPRETAÇÃO

Primeiramente devem ser analisadas as escalas das variáveis, a fim de determinar as operações que podem ser realizadas em cada uma delas. A tabela IX apresenta essas escalas.

Tabela IX – Escalas das variáveis.

<b>Variáveis</b>	<b>Nome</b>	<b>Escala</b>
Dependentes	Precisão	Razão
Independentes	Metodologia de Análise de Impacto	Nominal

### 5.4.1 Análise Tabular e Gráfica

O experimento obteve como resultado os dados apresentados na tabela X, onde os participantes que realizaram a metodologia de análise de impacto baseada em ontologia foram representados por “C0x” e o que utilizaram a metodologia baseada em requisitos foram classificados por “R0x”. Para cada requisito de mudança proposto e alterado pelo participante foi calculada a Precisão, através da fórmula 14. O valor apresentado na tabela X corresponde a média da Precisão obtida por cada participante, em relação aos requisitos de mudanças propostos.

Tabela X – Tabulação dos valores brutos obtidos após a execução do experimento.

Abordagem	Participante	Precisão
Análise de Impacto por Conceitos	C01	1,00
Análise de Impacto por Conceitos	C02	1,00
Análise de Impacto por Conceitos	C03	1,00
Análise de Impacto por Conceitos	C04	0,75
Análise de Impacto por Conceitos	C05	0,25
Análise de Impacto por Conceitos	C06	0,50
Análise de Impacto por Requisitos	R01	0,13
Análise de Impacto por Requisitos	R02	0,35
Análise de Impacto por Requisitos	R03	0,32
Análise de Impacto por Requisitos	R04	0,42
Análise de Impacto por Requisitos	R05	0,25
Análise de Impacto por Requisitos	R06	0,23

A figura 83 representa o gráfico de barras com a precisão dos artefatos que tiveram a análise de impacto determinada e que foram modificados pelos participantes.

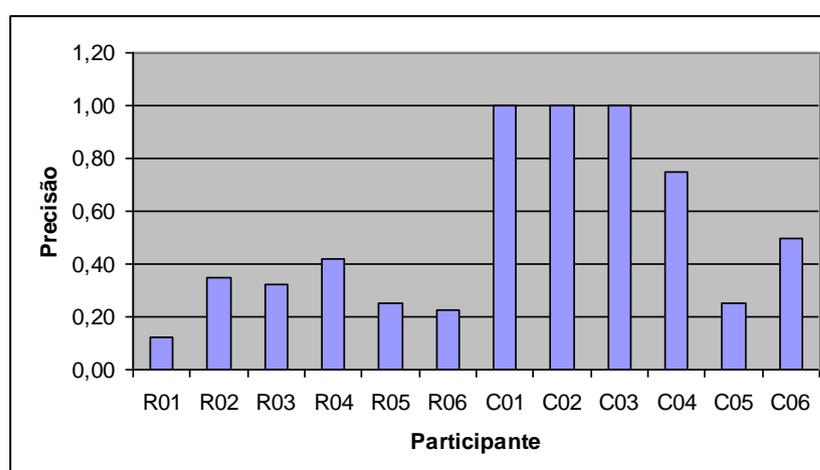


Figura 83 – Gráfico de barras relativo à precisão dos artefatos.

### 5.4.2 Estatística Descritiva

A variável precisão está caracterizada na escala razão. Isso permite o cálculo da normalidade e homocedasticidade, necessária para definir o tipo de teste das hipóteses (paramétrico ou não-paramétrico). Conforme definido no projeto do experimento, o padrão para tipo de teste previsto é o *Teste T* para duas amostras independentes, caso o teste empregado seja paramétrico, ou *Mann-Whitney*, caso seja não paramétrico.

Para avaliar a precisão, os dados tabulados durante o experimento serão caracterizados com o objetivo de visualizar tendências centrais e dispersões. Os dados que forem determinados como anormais ou incertos devem ser eliminados através da redução do intervalo de dados, pois esses distorcem a integridade da conclusão do experimento. Por último, será realizado o teste das hipóteses que compreende a avaliação estatística dos dados até certo nível de significância. O nível de significância adotado (*p-value*) para todos os testes é de 5%. O *p-value* compreende o menor nível de significância com que se pode rejeitar a hipótese nula.

### 5.4.3 Hipótese: Precisão

Para iniciar a validação da hipótese, é verificada a distribuição dos dados coletados. Para isso, é gerado o gráfico de dispersão *boxplot* para a identificação dos *outliers*. A figura 84 apresenta o gráfico.

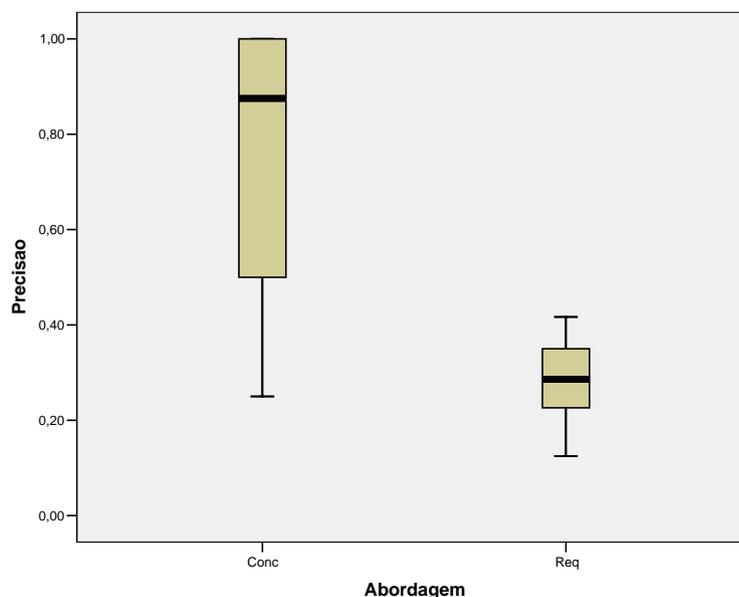


Figura 84 – Gráfico de dispersão para a variável precisão.

Como verificado na figura 84, a variável precisão possui *outliers* moderados, sendo assim, nenhum participante será extraído da amostra, já que não há risco de haver distorção. Pode-se notar no gráfico, previamente, que a dispersão é maior quando utilizada a análise de impacto baseada na rastreabilidade por conceitos, embora a mediana da variável precisão esteja perto de 100%. Os resultados para a análise de impacto baseada na rastreabilidade por requisitos são mais uniformes, mas menos precisos.

A próxima etapa consiste em identificar se os dados seguem uma distribuição normal. Para se avaliar a normalidade, é definida uma hipótese nula e uma hipótese alternativa, conforme:

- a)  $H_0$ : a distribuição é normal;
- b)  $H_1$ : a distribuição não é normal.

Existem duas formas para se avaliar a distribuição normal dos dados, que compreendem o *Teste de Kolmogorov-Smirnov* e o *Teste de Shapiro-Wilk*. O primeiro é utilizado para identificar a normalidade em variáveis com pelo menos 30 valores e o segundo em variáveis com menos de 50 valores. A tabela XI apresenta os testes de normalidades para a amostra utilizando o *Teste de Shapiro-Wilk*.

Tabela XI – Teste de normalidade *Shapiro-Wilk* para a variável precisão.

Variável	Abordagem	Estatística	Grau de Liberdade	Significância
Precisão	Análise de Impacto por Conceitos	0,831	6	0,110
	Análise de Impacto por Requisitos	0,984	6	0,970

Com base na tabela XI, observa-se que a significância dos dados do *Teste de Shapiro-Wilk* é superior, em ambas as amostras (análise de impacto baseada na rastreabilidade por conceito e análise de impacto baseada na rastreabilidade por conceito), ao nível de significância definido (0,05 ou 5%). Com esta informação, não há indícios para rejeitar a hipótese nula sobre a distribuição da normalidade, conseguindo assim o primeiro requisito para utilização de teste paramétrico para duas amostras independentes.

O segundo requisito requer a análise da homocedasticidade, tornando necessário analisar a variância das duas amostras. Com este objetivo, definem-se duas hipóteses:

- a)  $H_0$ : As variâncias são iguais;
- b)  $H_1$ : As variâncias não são iguais.

O teste da hipótese acima é realizado com a significância obtida diretamente através do *Teste de Levene*. Este é usado para testar se  $k$  amostras têm a mesma variância. A tabela XII apresenta os resultados obtidos para este teste.

Tabela XII – Teste de *Levene* para igualdade das variâncias sobre a variável precisão.

		Significância
Precisão	Assumindo variâncias iguais	0,032
	Não assumindo variâncias iguais	

Com base na tabela XII, verifica-se que o nível de significância para variâncias iguais (0,032) é inferior ao nível de significância definido (0,05 ou 5%). Com esta informação, pode-se rejeitar a hipótese nula para variâncias, não sendo mais possível utilizar o teste paramétrico. O próximo passo é utilizar o *Teste de Mann-Whitney*, para duas amostras independentes, por se tratar de uma alternativa não-paramétrica para o *Teste T*.

O *Teste de Mann-Whitney* para duas amostras independentes é utilizado para comprovar se as diferenças entre as médias observadas nos dois grupos independentes são estatisticamente significativas. Com base na declaração das hipóteses, sugere-se:

- a)  $H_0$ : Não há diferença entre as médias ( $\mu_{air} = \mu_{aic}$ );
- b)  $H_1$ : Há diferença entre as médias ( $\mu_{air} \neq \mu_{aic}$ ).

Os resultados da aplicação do teste estão na tabela XIII.

Tabela XIII – Teste não paramétrico de *Mann-Whitney* para a variável precisão.

	U de Mann-Withney	W de Wilcoxon	Z	Sig. Assimpt. (bilateral)	Sig. Exata [2*(Sig.Unilateral)]
Precisão	3,500	24,500	-2,342	0,019	0,015 <sup>a</sup>

(a) Não corrigidos para os empates

Como o grau de significação associado (*Sig. Assimpt.*) é 0,019 e é menor que a significância assumida de 0,05, deve-se rejeitar  $H_0$ . Com a rejeição desta, comprova-se a hipótese  $H_1$ , que afirma que há diferença entre as médias, ou seja, existe diferença de média entre a análise de impacto baseada na rastreabilidade por conceitos e na análise de impacto baseada na rastreabilidade por requisitos.

Pela análise estatística dos dados, consegue-se recuperar duas informações:

- a) A distribuição da precisão não é normal, o que implica na execução de testes não paramétricos;
- b) Utilizando o *Teste de Mann-Whitney*, conseguiu-se verificar que existem diferenças entre as médias das duas amostras  $\mu_{air}$  e  $\mu_{aic}$ .

Utilizando o *Teste de Mann-Whitney*, conseguiu-se apenas rejeitar a hipótese nula, porém não foi possível validar as hipóteses alternativas, pois não é possível extrair relações de “maior do que” com o teste aplicado. Porém, sugere-se comparar a análise descritiva das médias da amostra, conforme a tabela XIV.

Tabela XIV – Estatística descritiva para a variável *precisão*.

<b>Abordagem</b>	<b>Média</b>
Análise de Impacto por Requisitos	0,281
Análise de Impacto por Conceitos	0,750

Observando as médias da variável precisão para cada uma das metodologias, pode-se verificar que, em média, a precisão da análise de impacto por conceitos é maior do que a precisão da análise de impacto por requisitos.

## 5.5 AVALIAÇÃO QUALITATIVA

Através do experimento foi possível avaliar quantitativamente as metodologias de análise de impacto. Para realizar a análise qualitativa, os participantes responderam um questionário ao término de suas atividades, conforme apresentado ao final desse apêndice. Como critério de resposta, foi utilizada a escala Likert que possui cinco pontos para o grau de satisfação. O objetivo de usar essa pesquisa de opinião era conhecer a percepção dos participantes quanto a usabilidade, utilidade e esforço para usar as diferentes metodologias.

As cinco questões propostas podem ser verificadas no Apêndice B. Os dados brutos obtidos como resultados estão apresentados na tabela XV, onde o participante identificado como “R0x” é referente a abordagem de análise de impacto baseada na rastreabilidade por requisitos e o participante “C0x” é referente a abordagem ontológica.

Tabela XV – Resultados da avaliação qualitativa.

<b>Participante</b>	<b>Q1</b>	<b>Q2</b>	<b>Q3</b>	<b>Q4</b>	<b>Q5</b>
R01	3	3	2	3	2
R02	3	3	4	2	3
R03	3	3	2	2	2
R04	2	3	2	4	2
R05	3	3	4	3	3
R06	4	3	4	4	3
C01	4	5	4	3	4
C02	4	5	4	4	4
C03	4	4	3	4	5
C04	3	4	2	3	3
C05	5	5	4	4	5
C06	4	4	3	4	4

A média aritmética para cada uma das questões é apresentada na tabela XVI.

Tabela XVI – Média da satisfação das questões sobre as abordagens.

<b>Abordagem</b>	<b>Q1</b>	<b>Q2</b>	<b>Q3</b>	<b>Q4</b>	<b>Q5</b>
Análise de Impacto por Requisitos	3,0	3,0	3,0	3,0	2,5
Análise de Impacto por Conceitos	4,0	4,5	3,3	3,7	4,2

Comparando-se as médias obtidas para cada uma das abordagens, pode-se perceber que as médias obtidas pela metodologia de análise de impacto baseada na rastreabilidade por conceitos foi maior para todas as questões. Verifica-se então que qualitativamente essa metodologia é melhor do que a metodologia de análise de impacto por requisitos, para esse conjunto de participantes.

## 5.6 CONSIDERAÇÕES

Neste capítulo foram apresentadas as avaliações quantitativas, através do uso da Engenharia de Software Experimental, e qualitativa, através de uma pesquisa de opinião. A intenção era comparar as metodologias de análise de impacto baseada por conceitos e por requisitos, sendo esta última proposta por O’Neal [ONE03].

O experimento fez uso da rastreabilidade ontológica, que está inserida dentro de um processo de desenvolvimento específico, o que não possibilita que essa experimentação seja generalizável a outros processos. Quanto ao contexto, foi analisado o impacto dos requisitos de mudança somente nas classes de projetos e as modificações foram solicitadas apenas no diagrama de classes. Decidiu-se restringir o experimento por questões de tempo para execução e escolheu-se esse artefato por representar as alterações estruturais no sistema.

Com os dados fornecidos pelos participantes, foi possível avaliar a variável precisão, a partir da qual seria feita a comparação das metodologias. Com a análise inicial da variável, percebeu-se que essa não apresentou uma distribuição normal das suas médias. Por esse motivo, usou-se o teste *Mann-Whitney* para avaliar as hipóteses. Esse teste comprova se as diferenças entre as médias observadas em dois grupos independentes são estatisticamente significativas, mas não estabelece se um grupo é superior ao outro. Uma vez que a aplicação do teste comprovou que as médias eram diferentes, foi necessário utilizar a análise descritiva das médias de cada metodologia e compará-las. Assim, verificou-se que a metodologia de análise de impacto baseada na rastreabilidade por conceitos tem uma precisão maior na determinação do conjunto de artefatos que serão impactados.

A última fase da experimentação é a Apresentação e Empacotamento do que foi realizado no experimento para que esse possa ser repetido. Sendo assim, é interessante a repetição deste para outros artefatos de software, tais como a descrição dos casos de uso, diagramas de atividades, de seqüência, ou quaisquer outros que se considerar relevante. A variável de precisão poderá ser analisada em outros contextos, possibilitando um maior conhecimento de como a metodologia de análise de impacto baseada na rastreabilidade por conceitos se comporta em universos distintos de projetos de manutenção e de participantes.

## 6 CONCLUSÃO E TRABALHOS FUTUROS

O objetivo dessa pesquisa era desenvolver uma metodologia de análise de impacto que fosse capaz de determinar com maior precisão os artefatos impactados por um requisito de mudança. Para tanto, buscou-se utilizar um tipo de rastreabilidade de artefatos que fizesse uso de semântica para tal. Isso porque se inferiu que a assertividade dos elementos rastreados que seriam afetados por uma alteração fosse maior. Encontrou-se então, na pesquisa que vinha sendo realizada por Noll[NOL05b], a integração de ontologias ao processo de desenvolvimento de software, onde uma das principais vantagens é a possibilidade de rastreamento dos artefatos.

Com essa realidade, decidiu-se propor uma metodologia de análise de impacto baseada na rastreabilidade através de ontologias. Para isso foi preciso conhecer como uma ontologia é estruturada, quais são seus elementos e como estes se relacionam. Igualmente foi necessário buscar na literatura outras propostas de análise de impacto, com a finalidade de encontrar os pontos fortes e fracos dessas, procurando determinar exatamente onde o uso da rastreabilidade semântica seria vantajoso.

Após o estudo desses e outros assuntos, iniciou-se o trabalho em direção a construção e estruturação da metodologia. Observou-se que não existiam na literatura trabalhos correlacionados que relacionassem modificações nos conceitos da ontologia a alterações sofridas por artefatos produzidos ao longo do desenvolvimento de software. Assim, foi necessário realizar um experimento exploratório para que se conhecesse como alterações na ontologia se refletiriam nos artefatos relacionados aos seus conceitos.

Ao analisar os resultados do experimento, foi possível determinar algumas influências, referentes a tipos de mudança e de relacionamento entre conceitos da ontologia. Percebeu-se que a mudança pode ser estrutural ou relacionada a regras de negócio e que cada uma delas afeta diferentemente os artefatos que são rastreados, dependendo da classificação do tipo de diagrama. Além disso, pode-se notar que artefatos que não estavam diretamente relacionados a um conceito poderiam ser afetados, uma vez que os conceitos da ontologia se

relacionam e uma alteração em um deles pode comprometer, de alguma forma, os outros aos quais está ligado. Para que o cálculo das probabilidades de impacto nos artefatos pudesse ser determinado, tem-se como premissa que essas variáveis sejam informadas na ontologia do sistema.

A primeira fase da metodologia é analisar o requisito de mudança e, após este ser modelado conceitualmente, gerar uma ontologia referente a esse requisito. Dessa forma, as duas ontologias, a do requisito de mudança e a do sistema, podem ser comparadas a fim de identificar os conceitos comuns as duas e saber, estruturalmente, quais estão sendo afetados e de que maneira. Com essas duas fases concluídas, podem-se rastrear os artefatos que estão relacionados aos conceitos que foram identificados. O último passo é calcular o impacto para todos os artefatos que foram encontrados, determinando assim, o conjunto dos artefatos com maior probabilidade de impacto.

Além do uso de ontologias para determinar o impacto de um requisito de mudança, outra contribuição dessa proposta foi o desenvolvimento da ferramenta ONTImpact. Com ela é possível informar as influências aos conceitos da ontologia, compará-las e realizar automaticamente os cálculos necessários, informando ao final os artefatos com maior probabilidade de serem modificados.

Alguns trabalhos futuros podem ser propostos para solucionar certas limitações dessa metodologia. Um deles é pesquisar a possibilidade de se identificar, também automaticamente, as alterações de regras de negócio, uma vez que hoje o usuário deve informá-la. Outro item que pode ser melhorado é inserir o uso de tesouros, realizando uma análise semântica dos conceitos e não morfossintática como é feita através de algoritmos de radicalização de palavras. Visando a evolução da metodologia, futuramente pode ser estudada uma forma de, utilizando as disciplinas de análise de impacto e rastreabilidade, garantir que a ontologia original do sistema em manutenção seja atualizada após a execução do requisito de mudança solicitado. Seria interessante também que essa metodologia, e sua ferramenta, fossem aplicadas em uma empresa, com um sistema efetivamente em manutenção. Desta forma, poderia-se estabelecer o uso real e prático do que foi proposto neste trabalho.

No que se refere as mudanças na estrutura da ontologia, deveriam ser analisadas as exclusões de conceitos e relacionamentos. Provavelmente, essas mudanças também exerceriam influência em artefatos relacionados a esses conceitos ou em outros a eles relacionados.

Quanto a ferramenta, suas funcionalidades poderiam ser adicionadas ao aplicativo ONTrace, proposto por Noll[NOL05b], já que as pesquisas de rastreabilidade ontológica, feita por ele, e a de análise de impacto, constante nesse trabalho, estão integradas. Dessa forma, o

usuário que queira calcular o impacto de uma mudança poderia realizar todas suas atividades em um só sistema.



## REFERÊNCIAS

- [AJI95] AJILA, S. **Software maintenance: an approach to impact analysis of objects change**. Software: practice and experience, v. 25, n.10, p. 1155-1181, Oct. 1995.
- [BAS94] BASILI, V. R, CALDIERA, G., ROMBACH, D. H. **The Goal Question Metric Approach**. Encyclopedia of Software Engineering - 2 Volume Set, pp 528-532, Copyright by John Wiley & Sons, Inc., 1994.
- [BAS96] BASILI, V. R. **The role of experimentation in software engineering: past, current and future**. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, ICSE, 18.,1996, Berlin. Proceedings... Berlin: Springer-Verlag, c1996.
- [BOE01] BOEHM, B. et al. **The Cocomo 2.0 software cost estimation model**. Estados Unidos, 2001. Disponível em: <<http://www.softwareengineer.org/>>. Acesso em: 15/04/2005
- [BOH96] BOHNER, S.; ARNOLD, R. **Software change impact analysis**. Los Alamitos, Calif.: IEEE Computer Society, c1996. 376 p.
- [BRO87] BROOKS, F. P. **No silver bullet: essence and accidents of software engineering**. Computer, v. 20, n. 4, p. 10-19, Apr. 1987.
- [CAS04] CASTOR, A. P. **Rastreamento de requisitos no processo de desenvolvimento de software orientado a agentes**. 2004. 128 f. Dissertação (Mestrado em Ciência da Computação) - Centro de Informática, Universidade Federal de Pernambuco, Recife, 2004.
- [CAR02] CARBONE, M; SANTUCCI, G. **Fast&&Serious: a UML based metric for effort estimation**. 2002. Trabalho apresentado na 16ª European Conference on Object-Oriented Programming - ECOOP, Málaga, 2002.
- [CAN05] CARNEIRO, R. E; BRITO, P. F. **Definição de uma ontologia em OWL para representação de conteúdos educacionais**. Trabalho apresentado no 7º Encontro de Estudantes de Informática do Tocantins - ECOINFO, Palmas, 2005.
- [CHI94] CHIDAMBER, S; KEMERER, C. **A metrics suite for object oriented design**. IEEE Transactions on Software Engineering, v. 20, n. 6, p. 476-493, June 1994.
- [COR03] CORRÊA, A. C. G. **Recuperação de documentos baseada em informação semântica no ambiente AMMO**. 2003. 93 f. Dissertação (Mestrado em Ciência da Computação) - Centro de Ciências Exatas e de Tecnologia, Universidade Federal de São Carlos, São Carlos, 2003.

[DIA04] DIAS, M. A. L. **Extração automática de palavras-chave na língua portuguesa aplicada a dissertações e teses da área das engenharias**. 2004. 138 f. Dissertação (Mestrado em Engenharia Elétrica) - Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, 2004.

[FAY85] FAY, D. S.; HOLME, G. D. **Help! I have to update an undocumented program**. In: CONFERENCE ON SOFTWARE MAINTENANCE, Proceedings... Washington, D.C.: IEEE Computer Society Press, c1985. p. 194-202.

[FEL03] FELICÍSSIMO, C. H. et al. **Geração de ontologias subsidiada pela engenharia de requisitos**. In: INTERNATIONAL WORKSHOP ON REQUIREMENTS ENGINEERING - WER, 6., 2003, Piracicaba. Proceedings... Piracicaba: [s.n.], 2003.

[FEN00] FENSEL, D. **Ontologies: a silver bullet for knowledge management and electronic commerce**. Berlin: Springer, 2001. 138 p.

[FET00] FENTON, N; NEIL, M. **Software metrics: roadmap**. In: CONFERENCE ON THE FUTURE OF SOFTWARE ENGINEERING - ICSE, 2000, Limerick, Ireland. Proceedings... New York: ACM Press, 2000. p. 357-370.

[GEL04] GELLER, J.; PERL, Y.; LEE, J. **Ontology challenges: a thumbnail historical perspective**. Knowledge and Information Systems, v. 6, n. 4, p. 375-379, July 2004.

[GEN03] GENERO, M., MIRANDA, D.; PIATTINI, M. **Defining metrics for UML state-chart diagrams in a methodological way**. In: CONCEPTUAL modeling for novel application domains. Berlin: Springer, c2003. p. 118-128. (Lecture Notes in Computer Science, 2814).

[GOM01] GOMES, P.; BENTO, C. **A case similarity metric for software reuse and design**. AI EDAM, v. 15, n. 1, p. 21-35, Jan. 2001.

[GON03] GONZALEZ, M.; LIMA, V. L. S. **Recuperação de informação e processamento da linguagem natural**. In: JORNADA DE MINI-CURSOS DE INTELIGÊNCIA ARTIFICIAL, 3., 2003, Campinas. Campinas: SBC, 2003. v. 3, p. 347-395. Realizada no 23º Congresso da Sociedade Brasileira de Computação.

[GOT94] GOTEL, O.C.Z.; FINKELSTEIN, C.W. **An analysis of the requirements traceability problem**. In: INTERNATIONAL CONFERENCE ON REQUIREMENTS ENGINEERING, 1., 1994. Proceedings.... Los Alamitos, Calif.: IEEE Computer Society Press, 1994. p. 94-101.

[GRU93] GRUBER, T. R. **A translation approach to portable ontology specifications**. Knowledge Acquisition, v. 5, n. 2, p. 199-220, 1993.

[IEE98] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. **Standard for software maintenance**: IEEE std 1219. New York, 1998. 52 p.

[JAV06] JAVA. **Java Technology**. Estados Unidos, 2006. Disponível em: <<http://java.sun.com/>>. Acesso em: 14/10/2006

[JEN06] JENA. **A Semantic Web Framework for Java**. Estados Unidos, 2006. Disponível em: <<http://jena.sourceforge.net/>>. Acesso em: 22/10/2006

- [KOG02] KOGUT, P. et al. **UML for ontology development**. The Knowledge Engineering Review, v. 17, n. 1, p. 61-64, 2002.
- [KIM02] KIM, H.; BOLDYREFF, C. **Developing software metrics applicable to UML models**, 2002. Trabalho apresentado na 16ª European Conference on Object-Oriented Programming - ECOOP, Málaga, 2002.
- [LOR94] LORENZ, M.; KIDD, J. **Object-oriented software metrics: a practical guide**. Englewood Cliffs, N.J.: Prentice Hall, 1994.
- [MAR98] MARCHESI, M. **OOA metrics for the unified modeling language**. In: EURO-MICRO CONFERENCE ON SOFTWARE MAINTENANCE AND REENGINEERING, 2., 1998, Florence. Proceedings...,1998. p. 67-73.
- [MIL99] MILLS, E. E. **Metrics in the software engineering curriculum**. Annals of Software Engineering, v. 6 n. 1-4, p. 181-200, 1998.
- [MQU06] McQUILLAN, J. A.; POWER, J. F. **Some observations on the application of software metrics to UML models**. In: ACM/IEEE INTERNATIONAL CONFERENCE ON MODEL DRIVEN ENGINEERING LANGUAGES AND SYSTEMS - MoDELS, 9., 2006, Genova. Proceedings...
- [NOL05a] NOLL, R. **Ontologias no processo de desenvolvimento de software**. Trabalho individual I, Instituto de Informática, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2005.
- [NOL05b] NOLL, R. **Integração de ontologias no processo de desenvolvimento de software**. Projeto de estudo e pesquisa. Instituto de Informática, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2005.
- [NOL05c] NOLL, R. **Integração de ontologias no processo de desenvolvimento de software**. Trabalho individual II, Instituto de Informática, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2005.
- [ONE03] O'NEAL, J. S. **Analyzing the impact of changing software requirements: a traceability-based methodology**. 2003. Tese de Doutorado - Faculty of the Louisiana State University, 2003.
- [OLI01] OLIVEIRA, D. V. **JDBC – Java Database Connectivity**. Videira: Universidade do Oeste de Santa Catarina, 2001. p.1-6.
- [PAL03] PALO, M. **Requirements traceability**. Helsinki: University of Helsinki, Department of Computer Science, 2003. 10p. Seminar report.
- [PEA96] PEARSON, J. K. **Requirements, traceability and formal software development or a further analysis of requirement traceability**. 1996.
- [PRE95] PRESSMAN, R. S. **Engenharia de software**. São Paulo: Makron Books, 1995. 1056 p.

- [PIN02] PINTO, M. B.; SACCOL, D. B. **Uso de ontologias na definição de esquemas para banco de dados semi-estruturados**. Trabalho apresentado na 3ª Jornada de Iniciação Científica do CEULP, 2002.
- [RAM01] RAMESH, B.; JARKE, M. Toward reference models for requirements traceability. **IEEE Transactions on Software Engineering**, v. 27, n. 1, p. 58-93, Jan. 2001.
- [SCH99] SCHACH, S. R. **Object-oriented and classical software engineering**. WCB, McGraw Hill, 1999.
- [SAL68] SALTON, G. **Automatic information organization and retrieval**. New York: McGraw-Hill, c1968. 514 p. (Computer Science Series).
- [SCO04] SCOTTO, M. et al. A relational approach to software metrics. In: ACM SYMPOSIUM ON APPLIED COMPUTING, 2004, Nicosia, Cyprus. **Proceedings...** New York: ACM Press, 2004. p. 1536-1540.
- [SOM03] SOMMERVILLE, I. **Software engineering**. Addison-Wesley, 2003.
- [TOR99] TORANZO, M.; CASTRO, J. A. **Comprehensive traceability model to support the design of interactive systems**. In: WISDOM'99 - INTERNATIONAL WORKSHOP ON INTERACTIVE SYSTEM DEVELOPMENT AND OBJECT MODELS, 1999, Lisboa. **Proceedings...** 1999. p. 1-20.
- [TRA02] TRAVASSOS, G. H.; GUROV, D.; AMARAL, E. A. G. do. **Introdução à engenharia de software experimental**. Rio de Janeiro: COPPE/UFRJ, 2002. 52 p. (RT-ES-590/02).
- [WMS04] WORKSHOP DE MANUTENÇÃO DE SOFTWARE MODERNA - WMSWM, 1., 2004, Brasília. **Anais eletrônicos...** Disponível em: <<http://www.ucb.br/ucbtic/wmswm-04/>>. Acesso em: 17 jul. 2006.
- [WOH00] Wohlin, C. et al. **Experimentation in software engineering: an introduction**. Boston: Kluwer Academic, 2000. 204 p.

## APÊNDICE A – RESULTADOS DO EXPERIMENTO

Neste apêndice é apresentado o experimento realizado com o objetivo de mapear as alterações ocorridas na ontologia do sistema, a partir de alguns requisitos de mudança para este. Para realizá-lo, foi usado um projeto pré-existente, desenvolvido por alunos do curso de Sistemas de Informação da Faculdade de Informática da PUCRS, para se obter os resultados. Esse projeto é denominado *IsGym* e tem como objetivo integrar todas as funcionalidades administrativas de uma academia de ginástica em um único sistema de informação. Nele, são executadas funções como manter cadastro de clientes, funcionários, fornecedores, recursos e utilização destes, assim como as fichas de acompanhamento das avaliações (biometria) e atividades dos alunos.

Esse projeto foi escolhido por duas características principais. A primeira é possuir uma rica documentação sobre a modelagem do sistema, seguindo a metodologia RUP e utilizando os principais diagramas indicados por esta. A segunda é por ter sido desenvolvido sem a intenção de ser aplicado nesse experimento, o que garante que seus dados não foram manipulados para algum resultado específico.

O negócio do sistema foi amplamente estudado, usando para esse fim modelos como Casos de Uso de Negócio, Atividades de Negócio, Interação de Negócio e Objetos de Negócio. A modelagem do ambiente foi feita para os contextos Efetuar Compra, Criar Ficha de Acompanhamento de Cliente e Gerar Gráfico de Desempenho. Como exemplo, as figuras 85, 86, 87 e 88 mostram, respectivamente, os modelos citados, para o segundo contexto.



Figura 85 – Modelo de caso de uso de negócio para o contexto Criar Ficha de Acompanhamento de Cliente.

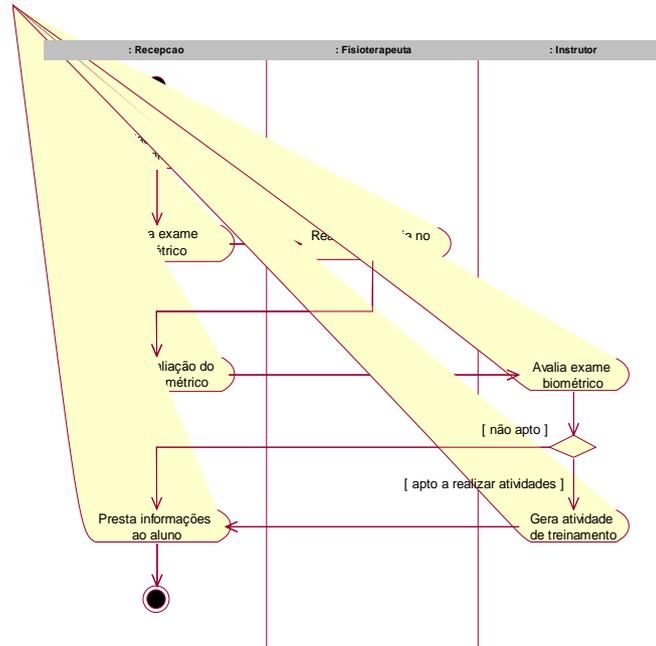


Figura 86 – Modelo de atividades de negócio para o contexto Criar Ficha de Acompanhamento de Cliente.

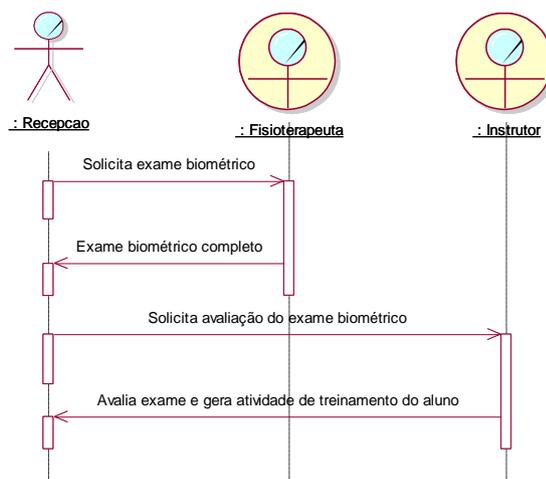


Figura 87 – Modelo de interação de negócio para o contexto Criar Ficha de Acompanhamento de Cliente.

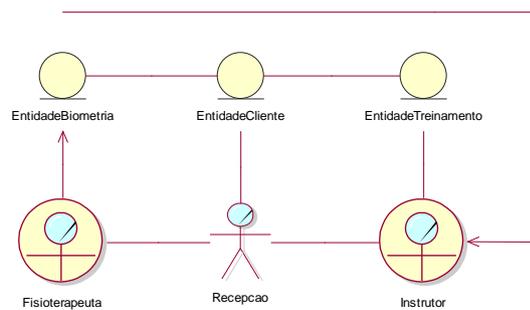


Figura 88 – Modelo de objetos de negócio para o contexto Criar Ficha de Acompanhamento de Cliente.



<b>Identificação</b>	UC3.	
<b>Caso de Uso</b>	Manter ficha de acompanhamento de cliente.	
<b>Ator</b>	Funcionário.	
<b>Pré-Condição</b>	Funcionário estar autenticado no sistema.	
<b>Pós-Condição</b>	Ficha pessoal criada.	
<b>Cenário</b>	Criar ficha pessoal	
<b>3.3.1. Criar ficha pessoal</b>		
Seqüência típica de eventos		
<b>Ação do ator</b>	<b>Resposta do sistema</b>	
<b>1</b>	Este caso de uso começa quando o Funcionário seleciona a opção <i>Criar ficha pessoal</i> na seção <i>Manter ficha de acompanhamento de cliente</i> do sistema.	
		<b>2</b> Sistema apresenta formulário para criação da ficha pessoal.
<b>3</b>	Funcionário informa nome, CPF, endereço, telefone, cidade, estado (uf) e código de acesso e senha de acesso do novo cliente.	
		<b>4</b> Sistema verifica se nome, CPF, endereço, cidade, estado (uf), código de acesso e senha de acesso são nulos, se CPF são numéricos não negativo e se CPF e código de acesso já existem no cadastro de fichas pessoais do sistema.
		<b>5</b> Sistema cadastra ficha pessoal com status "ativa" e confirma criação da ficha pessoal do cliente.
Seqüência alternativa de eventos		
<b>4a</b>	Se nome, CPF, endereço, cidade, estado (uf), código de acesso ou senha de acesso são nulos, ou se CPF não é numérico, ou se CPF é um numérico negativo, ou se CPF ou código de acesso já existem no cadastro de fichas pessoais do sistema: volta para 3.	

Figura 90 – Descrição do caso de uso Criar Ficha Pessoal.

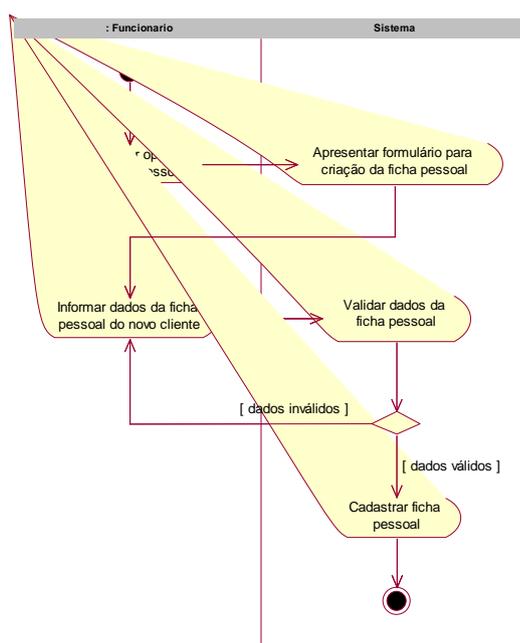


Figura 91 – Diagrama de atividades Criar Ficha Pessoal.

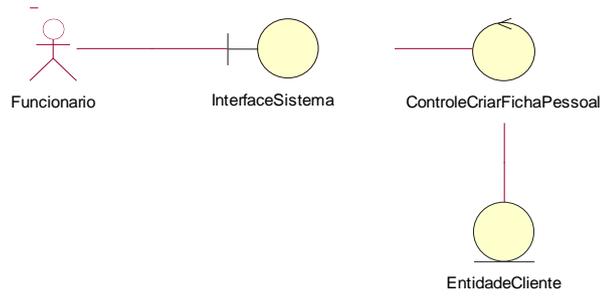


Figura 92 – Modelo de classe de análise Criar Ficha Pessoal.

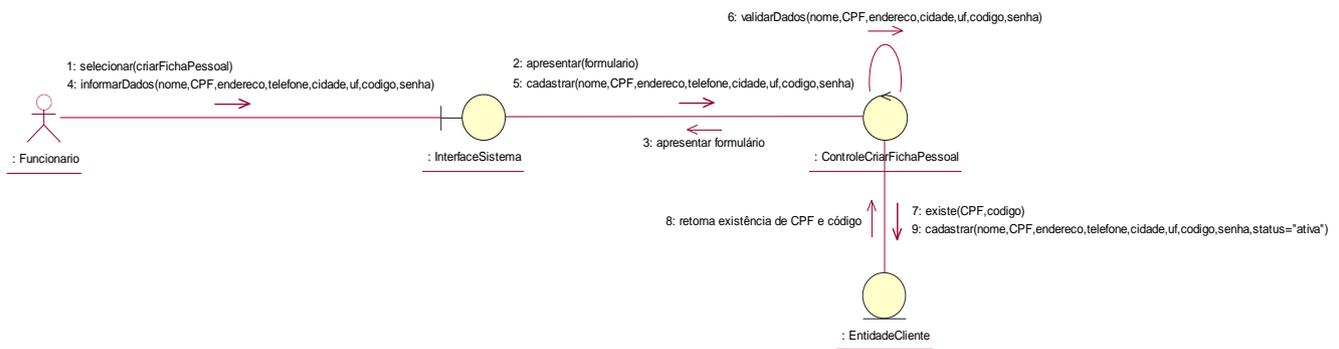


Figura 93 – Modelo de colaboração em análise Criar Ficha Pessoal.

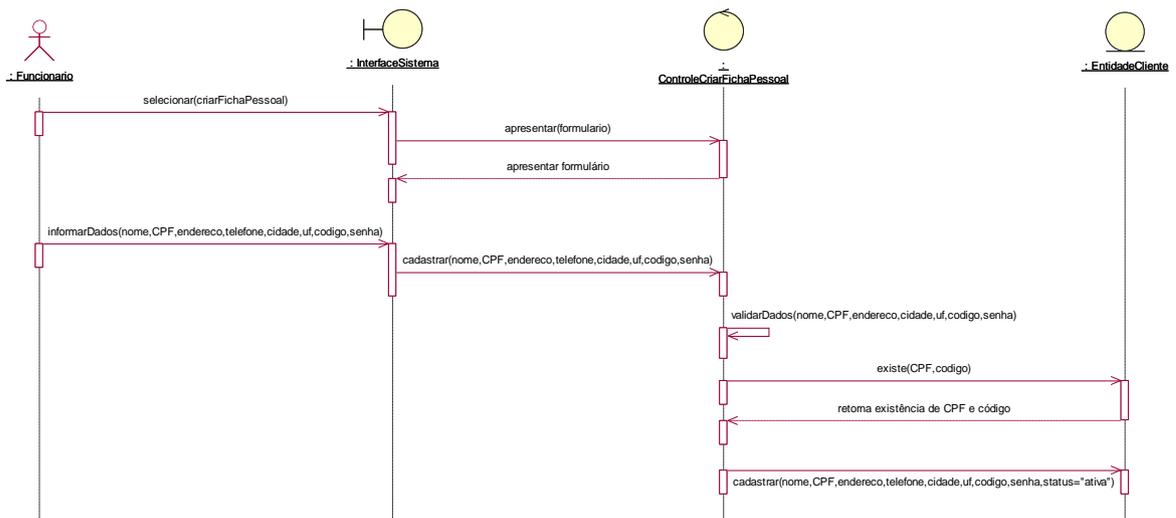


Figura 94 – Modelo de sequencia Criar Ficha Pessoal.



Figura 95 – Diagrama de classes do sistema *IsGym*.

Para aplicação do experimento foi escolhida a turma de Desenvolvimento de Sistemas Inteligentes do Programa de Pós-Graduação em Ciência da Computação da PUCRS. A execução do experimento foi planejada em duas fases. Na primeira, as áreas de análise de impacto e rastreabilidade semântica foram abordadas, numa aula explanatória sobre os assuntos. Nesse mesmo momento, a turma foi dividida em nove duplas e cada aluno respondeu um questionário para traçar o perfil dos participantes, que pode ser verificado na figura 96. Também foi explicado como o experimento seria conduzido, mostrando o que seria disponibiliza-

do e o que se exigiria como entrega. Para realização das tarefas, foi dado um prazo de duas semanas.

Questionário Perfil do Participante				
Formação				
Graduação	16			
Mestrado	15			
Doutorado	1			
Experiência Profissional (anos)				
	0 a 1	Mais de 1 a 3	Mais de 3	
Modelagem	6	3	6	
Implementação	5	4	6	
Rastreabilidade	12	1	2	
Proj. Manutenção	5	4	6	
Atuação				
<b>Empresa:</b>	<b>Cargo:</b>			
<b>Descrição:</b> Trabalham: 13 Estágio/Bolsa: 5; Programador: 1; Professor: 1; Líder Técnico: 1; Analista de Sistemas: 3; Engenheiro de SW: 2				
Conhecimento				
		Conhece mas nunca usou	Fez uso eventual	Uso corriqueiro
	Nenhum	Básico	Intermediário	Avançado
Ontologias	6	9	1	0
Processo Unificado	2	3	7	4
UML	0	3	3	10
ArgoUML	3	7	5	1
Domínio do Problema	1	6	2	2
Java/JSP	0	3	5	8
Access	0	1	12	3

Figura 96 – Resultado do questionário aplicado para conhecer o perfil dos alunos.

Com a análise dos dados dos alunos que responderam ao questionário, verifica-se que a maioria tem experiência acadêmica, sendo que a taxa de quem não tem experiência em modelagem, implementação e projetos de manutenção é semelhante a taxa daqueles que tem bastante experiência, sendo que muito poucos conheciam o assunto de rastreabilidade. Era interessante ter conhecimento dessas informações para ter-se uma noção da dificuldade, ou não, de realizar uma mudança na aplicação. Além dessas informações, foi solicitado que os alunos classificassem seus conhecimentos nas metodologias e ferramentas usadas no experimento. Quanto à ontologias, a turma ainda não possuía nenhum conhecimento nesse assunto, mas isso não foi impeditivo para a realização do experimento, no que diz respeito a análise de impacto da mudança, pois esta estava sendo usada somente para realizar a rastreabilidade dos artefatos com o uso da ferramenta ONTrace, fazendo parte da pesquisa e experimentação de Noll[NOL05c].

Para atingir a finalidade do experimento, alguns materiais foram fornecidos para os alunos. Dentre eles estavam a documentação completa do sistema, contendo os diagramas já citados e a descrição detalhada do negócio; o arquivo com a modelagem reproduzida no *ArgoUML*, uma vez que é nessa ferramenta que o ONTrace está integrado e é neste software que a ontologia do sistema pode ser gerada; o código fonte e a base de dados, além dos requisitos de mudança. Os documentos de entrega do experimento esperados eram um relatório das

alterações feitas, representando o conjunto de artefatos impactados, reproduzido na figura 97, o arquivo com a modelagem modificada, o código fonte e banco de dados alterados.

Conjuntos de Impacto					
Requisito de Mudança: 1				Grupo: x	
Artefatos					
Tipo I: Inclusão; E: Exclusão; A: Alteração			Classificação Im: Impactado; NI: Não-Impactado; N: Novo		
<b>Modelo de Domínio</b> Tempo: 15min					
Classe:	Livro	Tipo:	I	Descrição:	DataEmprestimo:Date
				Classif.:	NI
<b>Diagrama de Caso de Uso</b> Tempo: 20min					
Caso de Uso:	ReservarLivro	Tipo:	E	Descrição:	Não faz mais parte do escopo
				Classif.:	Im
<b>Descrição de Caso de Uso</b> Tempo: 10min					
Caso de Uso:	EmprestarLivro	Tipo:	A	Descrição:	Data de Emprestimo deve ser registrada
Caso de Uso:	ReservarLivro	Tipo:	E	Descrição:	Não faz mais parte do escopo
				Classif.:	Im
<b>Diagrama de Atividades</b> Tempo: 20min					
Atividade:	EmprestarLivro	Tipo:	A	Descrição:	Data de Emprestimo deve ser registrada
				Classif.:	Im
<b>Diagrama de Classes</b> Tempo: 20min					
Classe:	Livro	Tipo:	I	Descrição:	DataEmprestimo:Date
				Classif.:	NI
<b>Banco de Dados</b> Tempo: 10 min					
Caso de Uso:	EmprestarLivro	Tipo:	A	Descrição:	Data de Emprestimo deve ser registrada
				Classif.:	Im
<b>Código Fonte</b> Tempo: 1h					
Fonte:	EmprestarLivro.jsp	Tipo:	A	Descrição:	Comentar código fonte onde a alteração foi realizada. Seguir o padrão de comentário:
Fonte:		Tipo:		Descrição:	
Fonte:		Tipo:		Descrição:	
Fonte:		Tipo:		Descrição:	
Fonte:		Tipo:		Descrição:	
Fonte:		Tipo:		Descrição:	
				/*Requisito de Mudança nº:	
				Descrição da alteração: */	
Descrever de que forma os artefatos foram encontrados (o raciocínio utilizado):					

Figura 97 – Exemplo de preenchimento do relatório de conjunto de impacto.

O relatório de conjunto de impacto presente na figura 94 está preenchido com dados fictícios para demonstrar seu preenchimento. A finalidade era conhecer que tipo de raciocínio foi utilizado para encontrar os artefatos impactados, já que não foi sugerido o uso de uma metodologia para tal. Os artefatos que foram selecionados para serem analisados são os listados abaixo:

- Modelo de Domínio: era interessante analisar alguma alteração nesse diagrama, já que a ontologia é gerada a partir dele;
- Diagrama de Caso de Uso: foi selecionado pois a alteração poderia ser realizada modelando um novo caso de uso;
- Descrição de Caso de Uso: uma vez que foram propostas alterações nas regras de negócio, essas deveriam ser informadas na descrição do caso de uso;
- Diagrama de Atividades: com a alteração de regras de negócio, provavelmente novas atividades seriam necessárias na modelagem anteriormente existente;
- Diagrama de Classes: analisado para verificar se houve mudança de tipo de atributo, inclusão de novo atributo ou classe;
- Banco de Dados: embora as alterações no banco de dados não possam ser rastreadas pela ferramenta ONTrace, era interessantes saber como as alterações se refletiam nele;
- Código Fonte: também não é possível rastrear o trecho do código onde deve ser feita a alteração, mas era necessário saber se as alterações feitas na modelagem seriam refletidas neste. Além disso, quando não se utiliza uma metodo-

logia de rastreabilidade, artefatos que precisam ser alterados podem ser descobertos somente quando o código é alterado.

Outras duas informações constantes no relatório são: o tipo de mudança que estava sendo feita e a classificação do artefato. A primeira refere-se à inclusão, exclusão ou alteração de elementos dos diagramas ou do código. A segunda informação foi baseada na proposta de Bohner[BOH02], que separa os artefatos em conjuntos diferentes a medida que esses são encontrados. No caso, tentou-se perceber quais artefatos foram rastreados primeiramente e desses, quais realmente foram impactados (Im:Impactado), quais não foram impactados (NI:Não Impactado) e os que foram encontrados depois (N:Novo), possivelmente quando o código fonte estava sendo alterado.

Com finalidades distintas, foram criados três requisitos de mudança. O primeiro deles foi citado no capítulo 4 e já foi amplamente abordado. O segundo requisito de mudança visava modificar a semântica e o *range* de conceitos da ontologia. O terceiro requisito alterava uma regra de negócio. As figuras 98 e 99 apresentam, respectivamente, essas solicitações.

Visando ter uma carta de fornecedores confiáveis, a GymJoe decidiu classificar seus fornecedores pela qualidade do item vendido ao invés de classificar pelo prazo de entrega. Esta decisão foi tomada, pois mesmo havendo atraso de alguns fornecedores, seus itens são superiores aos da concorrência. Quando os itens forem entregues, o funcionário deve classificá-los em: bom e ruim, não havendo mais nenhum outro critério válido para classificação de fornecedores.

Figura 98 – Requisito de mudança 2 para sistema *IsGym*.

Foi identificado um problema na promoção de descontos da mensalidade, já que a adesão foi muito grande. Os clientes estavam determinando objetivos fáceis de serem atingidos. Consequentemente, o número de clientes com desconto máximo cresceu consideravelmente. Para que isso não ocorra, o sistema deve calcular, a partir do peso, altura, sexo e idade do cliente, a taxa de gordura ideal que deve ser atingida. Se o objetivo for atingido, será concedido um desconto de 10%. Os critérios são:

- Se o cliente está na faixa “Excesso de gordura”, o objetivo deve ser passar para faixa “Moderado”.
- Se o cliente está na faixa “Moderado”, o objetivo deve ser passar para “Ideal”.
- Se o cliente está na faixa “Ideal”, o objetivo deve ser permanecer nessa faixa.
- Se o cliente está na faixa “Baixo”, o objetivo deve passar para faixa “Ideal”.
- Se o cliente está na faixa “Excepcionalmente baixo”, o objetivo deve ser passar para faixa “Baixo”.

#### Cálculo da Taxa de Gordura:

Sexo = 1(homem); 0(mulher)

$$\text{Imc} = \text{peso} / ((\text{altura}_m * 100 + \text{altura}_{cm}) / 100)^2$$

$$\text{igc} = (1.2 * \text{imc}) + (0.23 * \text{idade}) - (10.8 * \text{sexo}) - 5.4;$$

Classificação	Homens	Mulheres
excepcionalmente baixo	6% a 10%	10% a 15%
baixo	11% a 14%	16% a 19%
ideal	15% a 18%	20% a 25%
moderado	19% a 24%	26% a 29%
excesso de gordura	maior que 25%	maior que 30%

Figura 99 – Requisito de mudança 3 para sistema *IsGym*.

Com a análise dos documentos entregues, percebeu-se que houve dificuldade em usar a classificação de impacto do artefato proposta. Possivelmente um dos motivos para isso ter ocorrido é que o experimento não foi realizado presencialmente, o que impediu o esclarecimento de dúvidas. Da mesma forma, a qualidade da informação do tipo de mudança que estava sendo realizada não foi positiva, não permitindo uma análise nesse sentido.

Sobre o raciocínio utilizado para encontrar os artefatos impactados, a técnica mais utilizada foi analisar os documentos de especificação do *IsGym* que foram entregues, verificando os diversos diagramas e modelos. Alguns grupos iniciaram a análise diretamente pelo código fonte, procurando nele por palavras-chave retiradas do requisito de mudança e após mantendo a integridade na especificação.

Os artefatos impactados pelo requisito de mudança 1, estão na figura 100, os impactados pelo requisito de mudança 2 estão presentes na figura 101, e pelo requisito de mudança 3, na figura 103. Pode-se perceber que as alterações não formam uniformes, pois sem o uso de uma metodologia de análise de impacto, que irá indicar os artefatos com maior probabilidade de serem afetados, as alterações são bastante dependentes da interpretação de quem a está desenvolvendo e do entendimento que se tem do negócio e da estrutura do sistema.

Diagrama	Artefato	Percentual Alterações/Grupo
Modelo de Dominio	FichaBiometrica	89%
	Mensalidade	56%
Diagrama de Caso de Uso		
Descrição de Caso de Uso	GerarMensalidade	89%
	CriarFichaBiometrica	78%
	AlterarUltimaFichadeBiometria	78%
	AcessarFichadeBiometria	67%
		0%
Diagrama de Atividade	CriarFichaBiometria	56%
	GerarMensalidade	56%
	AlterarUltimaFichadeBiometria	33%
	AcessarFichadeBiometria	22%
Diagrama de Classe		
Diagrama de Classe	FichaBiometrica	89%
	Mensalidade	44%
Banco de Dados		
Banco de Dados	Biometria	89%
	Mensalidade	56%
Código Fonte		
Código Fonte	gerarMensalidadeCliente	67%
	gerarMensalidade	33%
	menuFuncionario	22%
	manterMeta	11%
	criarMeta	11%
	alterarBiometria	22%
	criarFichaBiometria	22%
	alterarBiometria	11%
	criarFichaPessoal	11%
	alterarFichaPessoal	11%

Figura 100 – Artefatos impactados pelo requisito de mudança 1.

Diagrama	Artefato	Percentual Alterações/Grupo
Modelo de Dominio	Fornecedor	44%
	PedidoCompra	44%
Diagrama de Caso de Uso		0%
Descrição de Caso de Uso	AlterarFornecedor	22%
	ClassificacaoFornecedor	89%
Diagrama de Atividade		
Diagrama de Atividade	ClassificarFornecedor	56%
Diagrama de Classe		
Diagrama de Classe	Fornecedor	78%
	PedidoCompra	56%
Banco de Dados		
Banco de Dados	Fornecedor	44%
	PedidoCompra	22%
Código Fonte		
Código Fonte	gravaClassificacao	78%
	classificarFornecedor	44%
	alterarFornecedor	11%
	manterFornecedor	11%

Figura 101 – Artefatos impactados pelo requisito de mudança 2.

Diagrama	Artefato	Percentual Alterações/Grupo
Modelo de Dominio	FichaBiometria	56%
	Cliente/Aluno	67%
Diagrama de Caso de Uso		0%
Descrição de Caso de Uso	GerarMensalidade	22%
	CriarFichaPessoal	78%
	CriarFichaBiometria	22%
	AlterarUltimaFichaBiometria	44%
	AcessarFichaPessoal	78%
	AcessarFichaBiometria	22%
Diagrama de Atividade	CalcularDesconto	11%
	CriarFichaPessoal	22%
	CriarFichaBiometria	11%
	AlterarUltimaFichaBiometria	22%
	AcessarFichaPessoal	22%
	AcessarFichaBiometria	11%
	GerarMensalidade	11%
Diagrama de Classe	FichaBiometria	89%
	Cliente	78%
	Mensalidade	67%
Banco de Dados	Biometria	67%
	Cliente	78%
Código Fonte	gerarMensalidadeCliente	44%
	gerarMensalidade	11%
	criarFichaPessoal	44%
	criarMeta	11%
	alterarFichaPessoal	33%
	alterarBiometria	33%
	criarFichaBiometria	33%
	alterarFichaBiometria	11%
incluirMeta	11%	

Figura 102 – Artefatos impactados pelo requisito de mudança 3.

O experimento foi bastante útil para identificar de que forma as alterações se refletiriam na ontologia e a probabilidade de impacto que cada uma delas teria. Assim, foi possível verificar que, quando se inclui um conceito ou esse é alterado, a influência da mudança é alta e que ela se nota estruturalmente na ontologia. Essa situação foi mais nitidamente verificada no requisito de mudança 2. Da mesma forma, quando regras de negócio são acrescentadas ou alteradas, casos dos requisitos de mudança 1 e 3, a influência dessas serão igualmente altas, mas mais fortemente nos diagramas dinâmicos. Além disso, foi possível notar a influência existente entre os relacionamentos dos conceitos, ou seja, quando artefatos são afetados, mas de forma indireta.

Esse experimento teve caráter exploratório, não sendo utilizado para validar a metodologia. Os resultados obtidos a partir dele, serviram apenas de insumo para que essa fosse estruturada.

## APÊNDICE B – INSTRUMENTAÇÃO DO EXPERIMENTO DE VALIDAÇÃO

Nesse apêndice é apresentado todo o material necessário para condução do experimento para validação da proposta. Primeiramente é apresentado o material necessário para realizar o experimento na abordagem da análise de impacto baseada na rastreabilidade ontológica. Após, o material para a análise de impacto baseada na rastreabilidade por requisitos. Por último é apresentado o questionário, que é comum às duas metodologias.

- Análise de Impacto baseada na rastreabilidade através de requisitos

<p><b>Atividade #01: Criação dos modelos conceituais e realizar a análise de impacto</b></p> <ol style="list-style-type: none"> <li>1. Ler os requisitos de mudança propostos para o sistema IsGym (RequisitosMudançasIsGym.doc)</li> <li>2. Fazer o modelo conceitual para cada requisito de mudança, no ArgoUML.</li> <li>3. Comparar os modelos conceituais com o modelo de domínio do sistema (isGym.zargo)</li> <li>4. Abrir a planilha (AIO.xls).</li> <li>5. Para cada classe comum aos modelos conceitual e de domínio, informar qual o tipo de alteração está sendo solicitado: estrutural (e suas subdivisões) ou de regra de negócio.</li> <li>6. Informar a influência que o conceito exerce sobre o artefato.</li> <li>7. Informar o valor da influência do tipo de diagrama, dependendo do tipo de mudança.</li> </ol>
<p><b>Atividade #02: Realizar a alteração no diagrama de classes</b></p> <ol style="list-style-type: none"> <li>1. Abrir o modelo isGym.zargo no ArgoUML.</li> <li>2. Modelar as alterações solicitadas nas classes necessárias.</li> <li>3. Entregar a planilha AIO.xls e o modelo isGym.zargo.</li> </ol>
<p><b>Atividade #03: Responder ao questionário</b></p>

Figura 103 – Tutorial fornecido para os participantes executarem a análise de impacto ontológica.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2	Requisito de Mudança	Artefato do tipo Classe	Tipo de Alteração	Ic	Id	Im								
3														
4														
5														
6	Requisito de Mudança A ou B	RN - Regra de Negócio												
7		Estrutural												
8		I - Inclusão da classe												
9		Ia - Inclusão de atributo na classe												
10		A - Alteração do tipo do atributo												
11		N - Nenhuma												
12														
13														
14														
15														
16														
17														
18														
19	<b>Rastreabilidade</b>													
20	<b>Conceito</b>	<b>Artefato do tipo Classe</b>												
21	Atividade	Atividade												
22	Cliente	Cliente												
23	FichaBiometrica	FichaBiometrica												
24	FichaTreinamento	FichaTreinamento												
25	Fornecedores	Fornecedores												
26	Funcionarios	Funcionarios												
27	Mensalidade	Mensalidade												
28	Orcamentos	Orcamentos												
29	PedidoCompra	PedidoCompra												
30														
31														

Rastreabilidade	
Conceito	Artefato do tipo Classe
Atividade	Atividade
Cliente	Cliente
FichaBiometrica	FichaBiometrica
FichaTreinamento	FichaTreinamento
Fornecedores	Fornecedores
Funcionarios	Funcionarios
Mensalidade	Mensalidade
Orcamentos	Orcamentos
PedidoCompra	PedidoCompra

Figura 104 – Planilha para preenchimento das variáveis da metodologia de análise de impacto ontológica.

- Análise de Impacto baseada na rastreabilidade através de requisitos

<p><b>Atividade #01: Definição das variáveis que fazem parte da metodologia</b></p> <ol style="list-style-type: none"> <li>8. Ler os requisitos de mudança propostos para o sistema IsGym (RequisitosMudançasGym.doc)</li> <li>9. Identificar a quais requisitos (ou casos de uso) os requisitos de mudança se referem.</li> <li>10. Abrir a planilha IAR.xls</li> <li>11. Para cada artefato (classe) rastreado para o requisito, informar a influência que o requisito exerce sobre ele.</li> <li>12. Para cada artefato (classe), informar a complexidade, esforço e a fase na qual este se encontra.</li> </ol>
<p><b>Atividade #02: Realizar a alteração no diagrama de classes</b></p> <ol style="list-style-type: none"> <li>4. Abrir o modelo isGym.zargo no ArgoUML.</li> <li>5. Modelar as alterações solicitadas nas classes necessárias.</li> <li>6. Entregar a planilha IAR.xls e o modelo isGym.zargo.</li> </ol>
<p><b>Atividade #03: Responder ao questionário</b></p>

Figura 105 – Tutorial fornecido para os participantes executarem a análise de impacto por requisitos.

	A	B	C	D	F	G	H	I	J
1	<b>Requisitos</b>		<b>Artefatos</b>	<i>Influência</i>		<b>Artefato</b>	<i>Complexidade e<sub>(c)</sub></i>	<i>Fase p<sub>(c)</sub></i>	<i>Esforço e<sub>(c)</sub></i>
3	Emitir Pedido de Compra	UC01	Funcionarios			<b>Classe</b>			
4			PedidoCompra			Atividade			
5			Fornecedores			Cliente	{baixa;média;alta}		
6	Efetuar Compra	UC02	Funcionarios			FichaBiometrica	{0,3;0,6;1}		
7			Orcamentos	{fraco;moderado;forte}		FichaTreinamento			
8			PedidoCompra	{0,3;0,6;1}		Fornecedores			
9			Atividade			Funcionarios	{Requisitos;Projeto;Implementação}		
10	Manter Ficha de Acompanhamento de Cliente	UC03	Cliente			Mensalidade	{1;2;3}		
11			FichaBiometrica			Orcamentos			
12			FichaTreinamento			PedidoCompra			
13			Funcionarios			<b>Requisitos (UC)</b>			
14			Atividade			Emitir Pedido de Compra			Medido em homem/hora
15	Acessar ficha de acompanhamento de cliente	UC04	Cliente			Efetuar Compra			
16			FichaBiometrica			Manter Ficha de Acompanhamento de Cliente			
17			FichaTreinamento			Acessar ficha de acompanhamento de cliente			
18			Funcionarios			Gerar gráficos de desempenho			
19			Atividade			Autenticar usuário			
20			Cliente			Manter funcionário			
21	Gerar gráficos de desempenho	UC05	FichaBiometrica			Manter fornecedor			
22			FichaTreinamento			Gerar Mensalidade			
23			Fornecedores						
24			Funcionarios						
25	Autenticar usuário	UC06	Cliente						
26			Funcionarios						
27	Manter funcionário	UC07	Funcionarios						
28			Fornecedores						
29	Manter fornecedor	UC08	Funcionarios						
30			Cliente						
31	Gerar Mensalidade	UC09	Funcionarios						
32			Mensalidade						
33									
34									
35									
36									
37									
38									

Figura 106 – Planilha para preenchimento das variáveis da metodologia de análise de impacto por requisitos.

▪ Questionário

<b>Avaliação Qualitativa da Proposta de Análise de Impacto</b>				
(1) Muito Ruim	(2) Ruim	(3) Regular	(4) Boa	(5) Muito Boa
<b>Q1</b>	Como você considera a usabilidade desta técnica de análise de impacto?			
<b>Q2</b>	Como você considera a utilidade desta técnica de análise de impacto?			
(1) Muito trabalhoso	(2) Pouco trabalhoso	(3) Regular	(4) Cômodo	(5) Muito Cômodo
<b>Q3</b>	Como você considera o esforço para aprendizagem da técnica?			
<b>Q4</b>	Como você considera o esforço para a definição das variáveis necessárias para determinar o impacto do artefato nessa metodologia?			
(1) Discordo Plenamente	(2) Discordo	(3) Não discordo nem concordo	(4) Concordo	(5) Concordo Plenamente
<b>Q5</b>	Você usaria na prática novamente esta metodologia de análise de impacto?			

Figura 107 – Questionário fornecido para os participantes.