

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

RAFAEL SANTOS DE ALMEIDA

**UMA FERRAMENTA SEMI-AUTOMATIZADA BASEADA EM
REQUISITOS NÃO-FUNCIONAIS PARA IDENTIFICAÇÃO DE
ASPECTOS EM FASE INICIAL DO DESENVOLVIMENTO DE
SISTEMAS**

TRABALHO DE CONCLUSÃO DE CURSO

PONTA GROSSA

2014

RAFAEL SANTOS DE ALMEIDA

**UMA FERRAMENTA SEMI-AUTOMATIZADA BASEADA EM
REQUISITOS NÃO-FUNCIONAIS PARA IDENTIFICAÇÃO DE
ASPECTOS EM FASE INICIAL DO DESENVOLVIMENTO DE
SISTEMAS**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Ciência da Computação, do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná.

Orientadora: Prof. Dra. Simone Nasser Matos

PONTA GROSSA

2014



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Ponta Grossa
Diretoria de Graduação e Educação Profissional



TERMO DE APROVAÇÃO

Uma Ferramenta Semi-Automatizada Baseada em Requisitos Não-Funcionais para Identificação de Aspectos em Fase Inicial do Desenvolvimento de Sistemas

por

RAFAEL SANTOS DE ALMEIDA

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 15 de julho de 2014 como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof^a Dr^a Simone Nasser Matos
Orientadora

Prof. Dr. Willian Massami Watanabe
Membro titular

Prof. Dr. Ionildo José Sanches
Responsável pelos Trabalhos
de Conclusão de Curso

Prof. MSc Vinicius Camargo Andrade
Membro titular

Prof. Dr Gleifer Vaz Alves
Coordenador do curso

- O Termo de Aprovação assinado encontra-se na Coordenação do Curso -

AGRADECIMENTOS

Agradeço a Deus pelo dom da vida e pela força que sempre me deu para prosseguir em minhas batalhas.

Agradeço a minha professora Simone Nasser Matos por todo auxílio, que mesmo em feriados, recessos, trabalhos, não deixou de me ajudar dedicando seu tempo e conhecimento para fazer com que esse trabalho pudesse ser desenvolvido.

Agradeço a minha família em especial meus pais Alvaro e Mariza que sempre foram meu porto seguro e minha fortaleza para minhas decisões e por todo apoio nesta jornada acadêmica.

Agradeço a minha namorada por todo apoio, carinho e paciência na distância e ausência física.

Agradeço aos professores por todo conhecimento e formação acadêmica.

Por fim agradeço aos meus amigos que sempre estiveram comigo independente da distância, e a todos aqueles que contribuíram para a realização deste trabalho.

“Se não podes entender, crê para
que entendas. A fé precede, o
intelecto segue” (Santo Agostinho)

RESUMO

ALMEIDA, Rafael Santos. **Uma Ferramenta Semi-Automatizada Baseada em Requisitos Não-Funcionais para Identificação de Aspectos em Fase Inicial do Desenvolvimento de Sistemas**. 2014. 81 f. Trabalho de Conclusão de Curso Bacharel em Ciência da Computação - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2014.

A Orientação a Aspectos tem como principal objetivo identificar o espalhamento de preocupações. Esse pode ser identificado na fase de análise ou de projeto. O processo de identificação nas fases iniciais é útil porque evita o espalhamento e duplicação do código. Existem na literatura vários métodos de levantamento de aspectos na fase de análise, os quais possuem seu processo de funcionamento. Alguns trabalham com temas, documentos de requisitos ou requisitos não-funcionais (RNF). Dentre esses se considera ideal para identificação de aspectos os baseados em RNF e que tenham uma ferramenta semi-automatizada para auxiliar no processo. Uma ferramenta semi-automatizada é aquela que contém um conjunto de passos capazes de ajudar um analista na inserção e controle dos dados. A ferramenta encontrada para detecção de aspecto foi a Minerando Aspectos em Requisitos, porém a mesma não consegue identificar todos os aspectos relacionados aos requisitos não-funcionais. Este trabalho desenvolveu uma ferramenta semi-automatizada fundamentada em um método que usa requisitos não-funcionais e seus respectivos atributos. O processo do método foi alterado, pois o mesmo não contemplava a detecção de aspectos entre sistemas e subsistemas. A entrada de dados para a ferramenta proposta são os: sistemas, subsistemas, casos de uso, cenários, fluxos, passos, requisitos não-funcionais e atributos dos RNF. A saída proporcionada pela ferramenta é a geração dos espalhamentos de aspectos entre sistemas, subsistemas e casos de uso. Os sistemas Framework de Preço de Venda e Controle de Obras foram cadastrados na ferramenta para testes. Por fim, realizou-se uma análise qualitativa entre as ferramentas da literatura e a proposta neste trabalho.

Palavras-chave: *Ferramenta semi-automatizada. Aspectos. Análise.*

ABSTRACT

ALMEIDA, Rafael Santos. **Uma Ferramenta Semi-Automatizada Baseada em Requisitos Não-Funcionais para Identificação de Aspectos em Fase Inicial do Desenvolvimento de Sistemas**. 2014. 81 f. Trabalho de Conclusão de Curso Bacharel em Ciência da Computação - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2014.

The Aspect Oriented aims to identify the scattered concerns. This can be identify in analysis or design phase. The identification process in the early stages is useful because it avoids code duplication and scattering. There are several methods in the literature of aspects identification in analysis phase, which have their operating process. Some work with themes, requirement document, or non-functional requirement (NFR). Among these is considered useful to aspect identification methods based in NFR and having a semi-automated tool to assist in the process. A semi-automated tool is one which contains a set of steps able to help an analyst in the insertion and data control. A tool found for aspect detection was Mining Aspect in Requirements, but the same can not identify all aspects related to NFR. This work developed a semi-automated tool based on a method that uses NFR and your attributes. The process of method was change, because the same not contemplate the aspect detection between systems and subsystems. The data entries for the proposed tool are: systems, subsystems, use case, scenario, flow, steps, NFR and NFR attributes. The output provided by the tool is the generation of scattering between systems, subsystems and use case. The systems Price Sale Framework and Works Control were registered in the tool for tests. A qualitative analysis between the tools of literature and the proposal been done to show their differences.

Key words: Semi-Automated. Aspect. Analysis.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de um candidato a aspecto: <i>Preencher Campos Obrigatórios</i>	17
Figura 2 – Exemplo de um Use Case Slice: Preencher Campos Obrigatórios..	18
Figura 3 – Exemplo de Crosscutting entre Casos de Uso	19
Figura 4 – Exemplo de Use Slice e Aspectos: Reservar Quarto	21
Figura 5 - Use Case Slice com Relacionamento <<include>>	21
Figura 6 - Use Case Slice com Relacionamento <<extend>>	22
Figura 7 - <i>Use Case Slice</i> com relacionamento do tipo generalização	22
Figura 8 - Tema na Fase de Análise	23
Figura 9 - Etapas do Método Baseado em Temas	25
Figura 10 - Exemplo de Relacionamento por vista	26
Figura 11 - Relacionamento Transversal entre Temas	27
Figura 12 - Exemplo de Vista de Temas Base	28
Figura 13 - Exemplo de Vista de Temas Aspecto	28
Figura 14 - Modelo de Identificação de Candidatos a Aspectos a partir de Requisitos.....	29
Figura 15 - Exemplo de Especificação de Preocupações: Tempo de Resposta.....	30
Figura 16 - Exemplo de Pontos de Vista de Preocupações em JPREView.....	31
Figura 17 - Processo do Método de Siqueira da Silva (2012)	33
Figura 18 - Espalhamento de Aspectos em Matrizes Adjacência.....	36
Figura 19 - Vista de Ação dos Requisitos do CMS.....	42
Figura 20 - Vista de Ação Clipped.....	43
Figura 21 – Processo do Método.	46
Figura 22 - Exemplo de Descrição Textual Logar no Sistema	48
Figura 23 – Exemplo de Espalhamento de Aspectos.....	50
Figura 24 – Exemplo de Matriz Auxiliar para o RNF Desempenho	50
Figura 25 - Diagrama de Classes da Ferramenta AspectNFR	54
Figura 26 – Modelo Entidade Relacionamento da Ferramenta AspectNFR.....	56
Figura 27 – Página Principal da Ferramenta AspectRNF.....	57
Figura 28 – Página Sistema	57
Figura 29 – Página de Cadastro de Sistema.....	58
Figura 30 – Página de Listagem de Sistema.....	58

Figura 31 – Página de Cadastro de Tipo de Fluxo	59
Figura 32 – Página de Validação de um Subsistema.....	59
Figura 33 – Página Subsistema	60
Figura 34 – Página de Cadastro de Subsistema.....	60
Figura 35 – Página de Cadastro de Atributo	61
Figura 36 – Página de Cadastro Caso de Uso.....	61
Figura 37 – Página de Listagem de Caso de Uso.....	62
Figura 38 – Página de Cadastro de Cenário	62
Figura 39 - Página de Cadastro Cenário com Passo	63
Figura 40 – Página para Vincular atributos de RNFs ao Passo	64
Figura 41 – Página de Listagem da Matriz por Caso de Uso	65
Figura 42 – Página de Visualização da Matriz de Adjacência “Logar no Sistema” – RNF – Desempenho.....	65
Figura 43 – Página Inicial do Espalhamento de Aspectos	66
Figura 44 – Página Inicial do Espalhamento de Aspectos Selecionando a Opção Caso de Uso.....	66
Figura 45 - Visualização Espalhamento de Aspectos por meio da Matriz Auxiliar.....	67
Figura 46 – Página Inicial para Cenário com Aspecto.....	68
Figura 47 – Cenário com Aspecto <i>Logar no Sistema</i> gerado pela ferramenta AspectRNF.....	68
Figura 48 – Página Inicial Aspectos Identificados por Requisito Não-Funcional.....	69
Figura 49 – Casos de Uso Vinculados a Atributos do RNF Desempenho	70
Figura 50 – Página com a Matriz de Adjacência para o Caso de Uso Realizar Cadastro de Obra.....	72
Figura 51 – Página de Seleção de Espalhamento de Aspectos entre Subsistemas.....	73
Figura 52 – Matriz Auxiliar entre os Subsistemas Controle de Login e Custos de Obra Civil.....	74
Quadro 1 - Mapeamento de Candidatos a Aspectos e suas Influências	32
Quadro 2 - Requisito Não-Funcional Desempenho e seus respectivos atributos.....	34
Quadro 3 - Modelo Matriz de Adjacência.....	34
Quadro 4 - Matriz de adjacência para a descrição textual Realizar Cadastro de Obra.....	35
Quadro 5 - Matriz auxiliar utilizada para tratar espalhamento de aspectos	36

Quadro 6 - Matriz auxiliar preenchida com aspectos do tipo desempenho	37
Quadro 7 - Descrição Textual Utilizada para Representar os Aspectos Identificados	38
Quadro 8 - <i>Poincut</i> Representando um Tipo de Aspecto	38
Quadro 9 – Análise dos Métodos	40
Quadro 10 – Análise das Ferramentas.....	45
Quadro 11 – Atributos Desempenho para a Descrição Logar no Sistema	48
Quadro 12 – Exemplo de aspectos dentro da descrição textual <i>Logar no Sistema</i>	52
Quadro 13 – Exemplo de <i>Poincut</i> para o Aspecto Usabilidade	53
Quadro 14 - Descrição Textual Realizar Cadastro de Obra	71
Quadro 15 - Matriz de Adjacência para a Descrição Textual Realizar Cadastro de Obra.....	72
Quadro 16 – Comparativo entre a Ferramenta Proposta e a Ferramenta Theme/Doc de Baniassad e Clarke (2004).....	75
Quadro 17 – Comparativo entre a Ferramenta Proposta e a Minerando Aspectos em Requisitos por Sampaio et al. (2005a).....	76

SUMÁRIO

1 INTRODUÇÃO	12
1.1 OBJETIVO.....	14
1.1.1 Objetivo Geral	14
1.1.2 Objetivos Específicos	14
1.2 ORGANIZAÇÃO DO TRABALHO	15
2 ABORDAGEM ORIENTADA A ASPECTOS	16
2.1 ANÁLISE ORIENTADA A ASPECTO	16
2.2 MÉTODOS DE IDENTIFICAÇÃO DE ASPECTOS NA FASE DE ANÁLISE.....	20
2.2.1 Identificação de Aspectos Baseada em Use Case Slice	20
2.2.2 Identificação de Aspectos Baseada em Temas.....	23
2.2.3 Identificação de Candidatos a Aspectos Baseado em Requisitos	29
2.2.4 Identificação de Aspectos Baseado em Atributos de Requisitos Não-Funcionais	32
2.3 ANÁLISE DOS MÉTODOS DE IDENTIFICAÇÃO DE ASPECTOS NA FASE DE ANÁLISE	39
2.4 FERRAMENTAS DE IDENTIFICAÇÃO DE ASPECTOS NA FASE DE ANÁLISE OU FERRAMENTAS SIMILARES.....	41
2.4.1 Theme/Doc.....	41
2.4.2 Minerando Aspectos em Requisitos	43
2.5 ANÁLISE DAS FERRAMENTAS DE IDENTIFICAÇÃO DE ASPECTOS NA FASE DE ANÁLISE	44
3 ASPECT-NFR: UMA FERRAMENTA PARA DETECÇÃO DE ASPECTOS 46	
3.1 PROCESSO DE IMPLEMENTAÇÃO DO MÉTODO DE SIQUEIRA DA SILVA	46
3.2 MODELAGEM DA FERRAMENTA	53
3.3 UM EXEMPLO DO USO DA FERRAMENTA PROPOSTA	57
4 TESTES E RESULTADOS	71
4.1 OUTROS EXEMPLOS DE APLICAÇÃO DA FERRAMENTA.....	71
4.2 ANÁLISE DAS FERRAMENTAS SIMILARES.....	74
5 CONCLUSÃO	78
5.1 TRABALHOS FUTUROS.....	79
REFERÊNCIAS	80

1 INTRODUÇÃO

A Programação Orientada a Objetos (POO) têm auxiliado e melhorado o desenvolvimento de software, uma vez que facilita o reuso de código, gerando ganhos de produtividade e na qualidade de software. Entretanto, a POO possui algumas limitações, tal como na abstração de Preocupações Transversais (*Crosscutting Concerns*), pois estes elementos costumam estar espalhados pelos módulos do sistema (SIQUEIRA DA SILVA, 2012).

Para suprir esta deficiência se desenvolveu a Programação Orientada a Aspectos (POA), paradigma este que tem como base a Orientação a Objetos e torna possível agrupar os requisitos segundo um critério estabelecido, separando os níveis de abstração durante o desenvolvimento do software. Além disto, identifica os possíveis aspectos do sistema. “Um aspecto é um tipo de preocupação a qual sua funcionalidade é desencadeada por outras preocupações e em múltiplas situações. Se a preocupação não for separada em um aspecto, esta funcionalidade terá de ser desencadeada explicitamente dentro do código relatado para a outra preocupação, entrelaçando-as. Adicionalmente, por causa de o desencadeamento estar em vários lugares, os desencadeamentos podem estar dispersos por todo o sistema” (CLARKE; BANIASSAD, 2005)

Identificar os aspectos nem sempre é uma tarefa simples, primeiramente por depender da fase a qual se encontra o projeto, pois quanto mais tarde ocorre sua identificação, mais entrelaçado podem se encontrar os módulos. Segundo ponto se refere à técnica utilizada para sua identificação, em que muitas destas são realizadas de forma manual pelo especialista (ROSENHAINER, 2004; SIQUEIRA DA SILVA, 2012).

Por estes motivos, diversos estudos e métodos têm sido desenvolvidos, tais como: Jacobson; NG (2004), Clarke; Baniassad (2005), Hornung (2010), Sampaio et al. (2005a) e Siqueira da Silva (2012), os quais enfatizam a detecção de aspectos na fase de análise. Ou seja, um bom projeto é aquele o qual possui uma fase de análise bem definida, pois é nesta etapa em que serão definidos os primeiros casos de uso, cenários do processo e se dá o início ao processo de definição dos Requisitos Funcionais (RN) e os Requisitos Não-Funcionais (RNF) (XAVIER,2009).

Dentre os métodos relatados, apenas o de Sampaio et al. (2005a) apresenta uma ferramenta de identificação semi-automatizada. Uma ferramenta semi-automática é aquela a qual apresenta passos desenvolvidos de forma automática pela ferramenta, software empregado e manual pelo auxílio de um analista na inserção e controle de dados movimentados pela ferramenta.

A ferramenta mineradora de Sampaio et al. (2005a) analisa qualquer tipo de documento textual, tais como: descrições de linguagem natural, descrições textuais de caso de uso e entrevista com o cliente. Além disso, utiliza-se de critérios como por exemplo substantivos ou verbos, repassando estas informações para um processador de linguagem natural denominado WMATRIX que se encarrega de processar os dados, retornando como resultado uma comparação das descrições fornecidas com o critério informado pelo analista. Por fim, a ferramenta filtra estes resultados e identifica os candidatos a aspectos.

Apesar de esta ferramenta possuir um processo semi-automático e utilizar um processador de linguagem natural, o autor relata que a mesma não é eficiente em identificar candidatos a aspectos a partir RNFs, os quais em muitos casos são tratados como aspectos. Outra deficiência é que o processador de linguagem natural WMATRIX está disponível apenas para a identificação de palavras da língua inglesa.

Na tentativa de sanar a limitação desta ferramenta, identificou-se na literatura que o método de Siqueira da Silva (2012) tem como foco levantar aspectos a partir dos RNFs. Embora este método não exija grande conhecimento sobre aspectos pelo analista, este requer atenção nas fases de montagem e preenchimento da matriz de adjacência e na identificação e avaliação de espalhamento de aspectos.

Este trabalho propõe uma ferramenta para o método desenvolvido por Siqueira da Silva (2012), denominada de AspectRNF, de forma a auxiliar o analista na construção da matriz de adjacência e na identificação de aspectos de forma semi-automatizada. No método de Siqueira da Silva (2012) não foi proposto à identificação de aspectos entre sistemas e subsistemas, porém a ferramenta proposta é capaz de realizar tal identificação.

A AspectRNF foi modelada usando um diagrama de classe e um modelo de entidade e relacionamento. A codificação foi feita usando a linguagem PHP 5.3, *JavaScript*, HTML e o MYSQL 5.5.24 para o banco de dados.

Para exemplificar o uso da ferramenta utilizou-se os exemplos apresentados no trabalho de Siqueira da Silva (2012) para o sistema Framemk (Framework para Formação de Preço de Venda) e Controle de Obras. Por fim, comparou-se a ferramenta com outras existentes na literatura para relatar suas diferenças no processo de identificação de aspectos.

1.1 OBJETIVO

A seguir serão descritos os objetivos gerais e específicos deste trabalho.

1.1.1 Objetivo Geral

Desenvolver uma ferramenta semi-automatizada capaz de identificar aspectos entre sistemas, subsistemas e casos de uso na fase de análise, utilizando como base cenários, requisitos não-funcionais e matriz de adjacência.

1.1.2 Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Identificar as características de ferramentas para identificação de aspectos existentes na literatura.
- Levantar os requisitos funcionais do método de Siqueira (2012), mostrando por meio de um processo as atividades essenciais do método.
- Modelar os requisitos identificados.
- Analisar qualitativamente a ferramenta proposta com as existentes.

1.2 ORGANIZAÇÃO DO TRABALHO

O presente trabalho dispõe de cinco capítulos. O Capítulo 1 relata a motivação e os objetivos para o desenvolvimento desta pesquisa.

O Capítulo 2 descreve sobre a abordagem orientada a aspectos, apresentado métodos e ferramentas de identificação de aspectos na fase de análise.

O Capítulo 3 mostra como o processo de como a ferramenta proposta funciona, além de apresentar sua modelagem e um exemplo de seu uso.

O Capítulo 4 narra um exemplo do uso da ferramenta usando dois sistemas e apresenta uma análise qualitativa entre as ferramentas de identificação de aspectos da literatura com a proposta neste trabalho.

Por fim, o último capítulo expõe as conclusões finais e as propostas para trabalhos futuros que podem ser desenvolvidos a partir desta pesquisa.

2 ABORDAGEM ORIENTADA A ASPECTOS

Este capítulo apresenta um apanhado geral sobre a Análise Orientada a Aspectos (AOA). A Seção 2.1 relata sobre os conceitos principais que envolvem a AOA. A Seção 2.2 descreve alguns métodos destinados à identificação de aspectos em fase inicial do desenvolvimento de software. A Seção 2.3 relata uma breve análise destes métodos. A Seção 2.4 apresenta ferramentas semi-automatizadas destinadas a identificação de aspectos na fase de análise. A Seção 2.5 relata uma análise das ferramentas semi-automatizadas para identificação de aspectos.

2.1 ANÁLISE ORIENTADA A ASPECTO

A fase de Análise se preocupa em investigar o domínio do problema desenvolvendo atividades que atendam as necessidades do cliente. Já a fase de projeto consiste na modelagem das atividades de criação, projetando-as para o programador, o qual transformará as informações da fase de projeto em algo implementável (LARMAN, 2007).

Segundo Clarke; Baniassad (2005, p. 6): “As fases de análise e projeto de um sistema são tão importantes quanto à implementação e muitos profissionais consideram estas fases as mais significantes em sua contribuição para o sucesso de um projeto como um todo”.

A Análise Orientada a Aspectos (AOA) trata nas primeiras atividades de criação de um software da identificação de aspectos a partir de Requisitos Funcionais e Não-Funcionais com base em descrições textuais e entrevistas com *Stakeholders*.

Requisitos Funcionais (RF) dizem respeito à regra de negócio e os Não-Funcionais (RNF) se relacionam com padrões de qualidade tais como: segurança, confiabilidade e desempenho (XAVIER, 2009).

Os RFs e RNFs podem ser traduzidos para aspectos, porém, os aspectos geralmente são identificados como RNFs, pois estes apresentam diversas similaridades, tais como: costumam estarem espalhados pelo sistema, estão relacionados a padrões de qualidade, sua identificação é importante para

o desenvolvimento de software, seu levantamento pode ser aplicado em sistemas em ambiente de produção (SIQUEIRA DA SILVA, 2012).

Identificados os RFs e RNFs se pode realizar a separação de preocupações (*Separation of Concerns*), em que se verifica também os pontos em que podem ocorrer das preocupações atravessarem uma a outra (*Crosscutting Concerns*) gerando o relacionamento entre preocupações do tipo *Crosscutting* (RASHID et al., 2002).

A AOA apresenta alguns conceitos os quais devem ser trabalhados, como por exemplo: Aspectos, Preocupações Transversais (*Crosscutting Concerns*), Separação de Preocupações (*Separation of Concerns*) e *Crosscutting*.

Aspectos são definidos como um tipo de preocupação na qual sua funcionalidade depende de outra preocupação. Caso esta funcionalidade não seja separada em um aspecto, esta deverá ser desencadeada explicitamente dentro do código, gerando o entrelaçamento entre preocupações. Adicionalmente, devido ao desencadeamento de funcionalidades estar em vários lugares, estes podem estar dispersos pelo sistema (CLARKE; BANIASSAD, 2005).

Supondo um sistema de *login*, onde há características que se repetem, como por exemplo, a validação do preenchimento de campos obrigatórios. Esta funcionalidade pode ser considerada um aspecto por ser usado em outras funcionalidades do sistema. A Figura 1 ilustra um exemplo do caso de uso candidato a aspecto *Preencher Campos Obrigatórios*, o qual apresenta como esteriótipo <<CA>> para a representação de que este caso de uso é um candidato a aspecto.

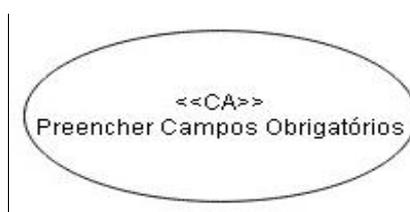


Figura 1 – Exemplo de um candidato a aspecto: *Preencher Campos Obrigatórios*

Fonte: Aatoria Própria

A Figura 2 ilustra aspectos dentro de um *Use Case Slice*. Um *Slice* contém pedaços de código de classes que são necessárias em várias partes do código. Ele é obtido a partir de realizações dos casos de uso, identificando as classes e seus métodos e posteriormente se estabelece quais métodos são os mais requeridos, sendo estes agrupados em um elemento, o aspecto (`<<aspecto>>`). Um aspecto pode ser composto de um ou mais pedaços de classes (KOSSOSKI, 2010).

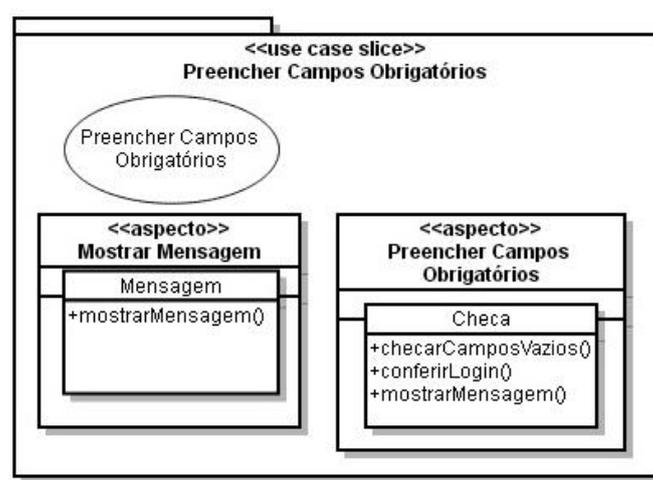


Figura 2 – Exemplo de um Use Case Slice: Preencher Campos Obrigatórios

Fonte: Kossoski (2010)

Esta figura ilustra um exemplo de *Use Case Slice* do preenchimento de campos obrigatórios, em que foram detectados dois tipos de aspectos *Mostrar Mensagem* e *Preencher Campos Obrigatórios* com suas classes específicas para sua realização *Mensagem* e *Checa*, respectivamente.

Outro conceito importante da AOA são as Preocupações Transversais (*Crosscutting Concerns*), que estão relacionadas às restrições globais de software e propriedades sistêmicas, tais como: segurança, desempenho, persistência e *logging*. Estes interesses costumam estar espalhados e/ou entrelaçados pelos módulos do sistema (HORNUNG, 2010). Um exemplo de preocupação transversal ocorre com o caso de uso *Preencher Campos Obrigatórios* o qual é atravessado pelos casos de uso *Logar no Sistema*, *Cadastrar Usuário*, *Enviar Mensagem* e *Mostrar Mensagem*.

A Separação de Preocupações (*Separation of Concerns*) propõe o encapsulamento de preocupações em aspectos, separando-os de forma a

tratar mais adequadamente as Preocupações Transversais (RASHID et al., 2002).

O relacionamento do tipo *Crosscutting* ocorre quando o desencadeamento de uma preocupação depende do comportamento de outra, gerando o entrelaçamento entre ambas (RASHID et al., 2002). A Figura 3 ilustra um exemplo de relacionamento por *Crosscutting* entre casos de uso.

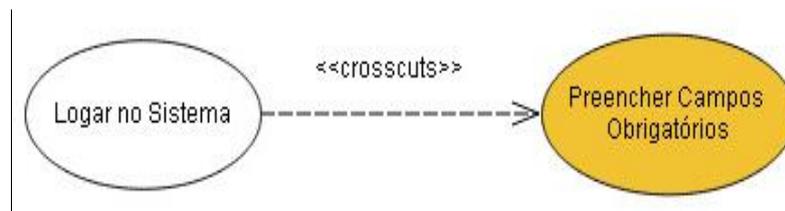


Figura 3 – Exemplo de Crosscutting entre Casos de Uso

Fonte: Kossoski (2010)

Nesta figura o caso de uso *Preencher Campos Obrigatórios* representa o elemento transversal, e o relacionamento do tipo *Crosscutting* é representado pelo estereótipo <<crosscuts>>.

Além dos conceitos explicados anteriormente da AOA, há outros relacionados a Programação Orientada a Aspectos (POA), tais como: *Join Point*, *Advice*, *Pointcut* e *Weaving* (SPRING FRAMEWORK, 2014), porém não serão abordados neste trabalho porque estão relacionados à fase de implementação.

De forma a maximizar os benefícios dos aspectos, a sua identificação deve ser aplicada nos estágios iniciais do desenvolvimento de software, como: na análise e engenharia de requisitos. A identificação de aspectos nos estágios iniciais auxilia a separação de preocupações na fase inicial do desenvolvimento de sistema, ao invés de deferir tais decisões em estágios posteriores, como: projeto e codificação, nos quais se devem realizar dispendiosas refatorações (SAMPAIO, 2005b).

Por este motivo, diversos estudos e métodos têm sido propostos para a identificação de aspectos logo nas primeiras fases do ciclo de desenvolvimento de um software em especial na fase de análise, são eles: Jacobson; NG (2004), Clarke; Baniassad (2005), Sampaio et al. (2005a) e Siqueira da Silva (2012). Esses métodos serão descritos na próxima seção.

2.2 MÉTODOS DE IDENTIFICAÇÃO DE ASPECTOS NA FASE DE ANÁLISE

Os métodos descritos a seguir foram selecionados por levantar aspectos em fase inicial do desenvolvimento de sistemas ou por apresentarem ferramentas semi-automáticas capazes de detectar aspectos na fase de análise.

2.2.1 Identificação de Aspectos Baseada em Use Case Slice

Casos de uso são utilizados para a identificação de preocupações de um sistema (JACOBSON; NG, 2004). Eles possuem uma limitação na orientação a aspectos, ou seja, não contém uma forma de encapsular os aspectos identificados. Desta forma, foi desenvolvida a unidade de encapsulamento denominada *Use Case Slice*.

Use Case Slice é uma unidade que preserva a modularidade dos casos de uso durante sua realização e contém os elementos necessários para a realização de um caso de uso. Ele é identificado na forma de um pacote contendo o estereótipo <<use case slice>> e é composto de uma colaboração a qual descreve a realização do caso de uso, as classes específicas para a sua realização e as extensões de classes existentes para a sua realização (JACOBSON; NG, 2004).

A Figura 4 ilustra um exemplo de *Use Case Slice* em que a colaboração é *Reservar Quarto*. O *ControladorReservaQuarto* é uma classe específica para a sua realização e possui um vínculo de dependência com *Reservar Quarto*. O aspecto identificado *Reservar Quarto* possui uma classe de extensão denominada *Quarto* com a presença do método *atualizarDisponibilidade* onde esse é utilizado apenas pelo *ReservarQuarto*.

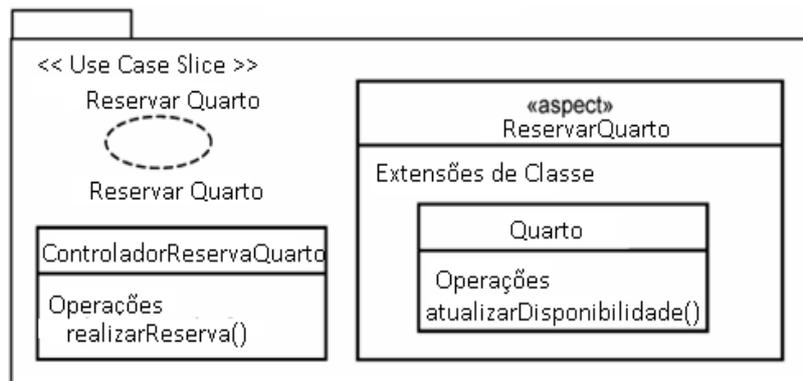


Figura 4 – Exemplo de Use Slice e Aspectos: Reservar Quarto

Fonte: Adaptado de Jacobson; NG (2004)

Os *Use Case Slices* possuem alguns tipos de relacionamento, tais como: *include*, *extend* e generalização. O relacionamento por *include* é modelado como uma dependência entre *Use Case Slices*. Na Figura 5 o *Use Case Slice ReservarQuarto* realiza a reserva do quarto. Para que este comportamento seja realizado é necessário que seja executado *Verificar Detalhes Quarto*. A verificação se torna obrigatória, uma vez que há o relacionamento do tipo *<<include>>*.

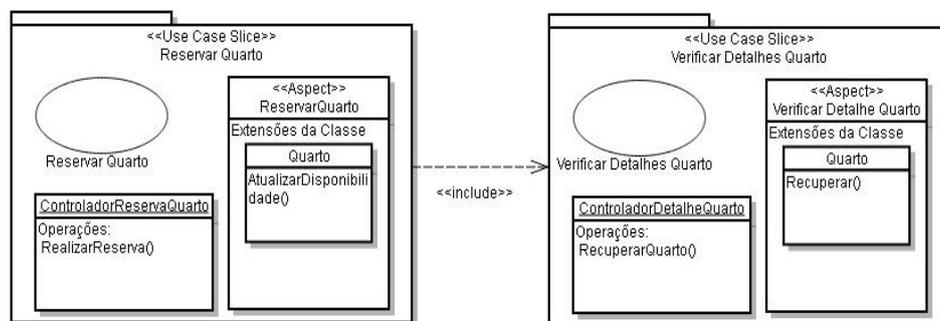


Figura 5 - Use Case Slice com Relacionamento <<include>>

Fonte: Adaptado de Jacobson; NG (2004)

O relacionamento por *extend* indica que um *Use Case Slice* poder ser acrescentado para descrever o comportamento de outro *Use Case Slice*, porém este relacionamento não é essencial.

A Figura 6 ilustra um exemplo de relacionamento por *extend*, em que a classe *Quarto* não é específica de apenas um caso de uso, não sendo definida dentro de um *Use Case Slice*. Desta forma, ela é definida dentro de uma

unidade representada pelo estereótipo *<<non-uc especific slices>>*, pois desta forma estará disponível para que outras classes possam estendê-la.

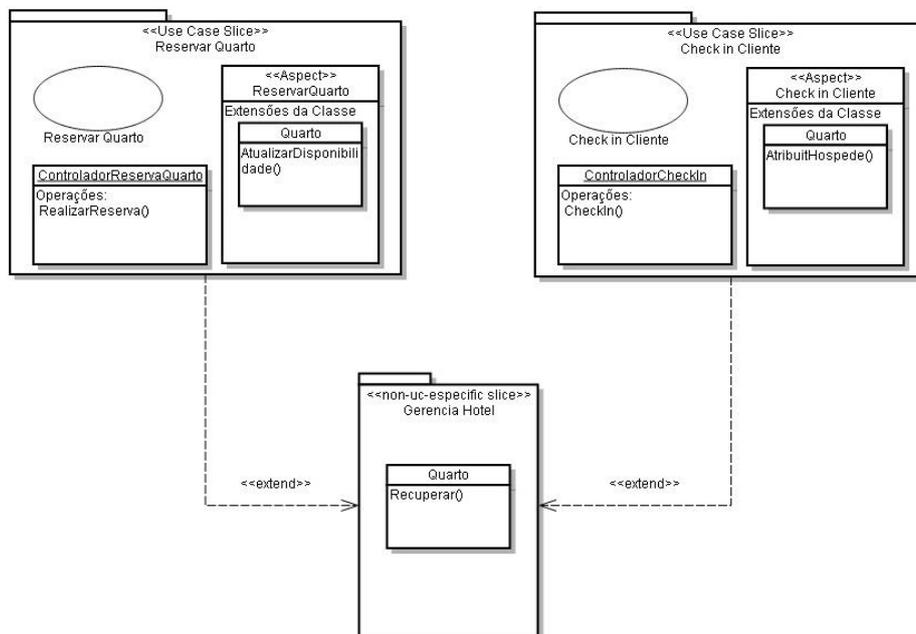


Figura 6 - Use Case Slice com Relacionamento <<extend>>

Fonte: Adaptado de Jacobson; NG (2004)

Outro relacionamento relevante entre *Use Case Slice* é o por generalização, em que há a presença de herança entre duas classes. A Figura 7 ilustra um exemplo de relacionamento entre do tipo generalização.

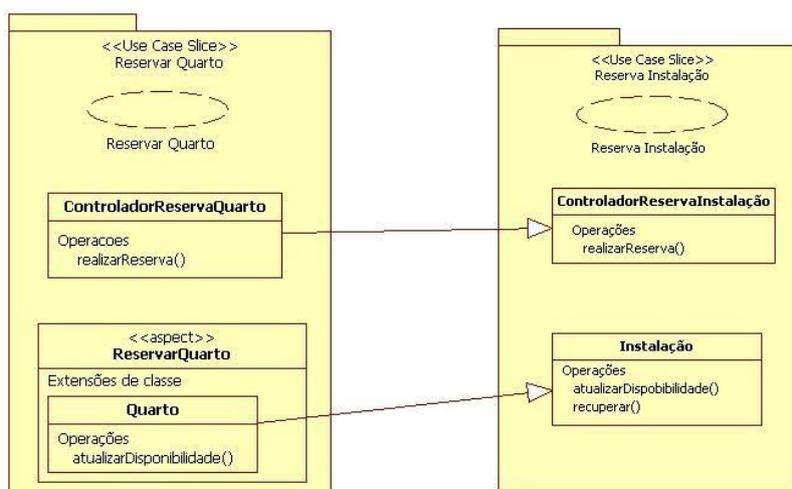


Figura 7 - Use Case Slice com relacionamento do tipo generalização

Fonte: Siqueira da Silva (2012)

Nesta figura a classe *ControladorReservaInstalação* apresenta o método *realizarReserva*, e a classe *ControladorReservaQuarto* herda este método e as características que a classe *ControladorReservaInstalação* possa apresentar. É possível visualizar também que a classe *Instalação* herda a operação *atualizarDisponibilidade* da classe *Quarto* dentro do aspecto *ReservarQuarto*.

2.2.2 Identificação de Aspectos Baseada em Temas

A abordagem baseada em temas proposta por Clarke; Baniassad (2005) propõe a identificação e o planejamento de cada tema do projeto separadamente, oferecendo uma forma de combiná-los posteriormente para formar um sistema completo.

Segundo Clarke; Baniassad (2005), temas são definidos como características ou preocupações presentes no sistema e podem ser identificadas tanto na fase de análise quanto na de projeto, sendo que na primeira os temas são mais abstratos, pois são participantes. A Figura 8 ilustra um exemplo de tema na fase de análise.

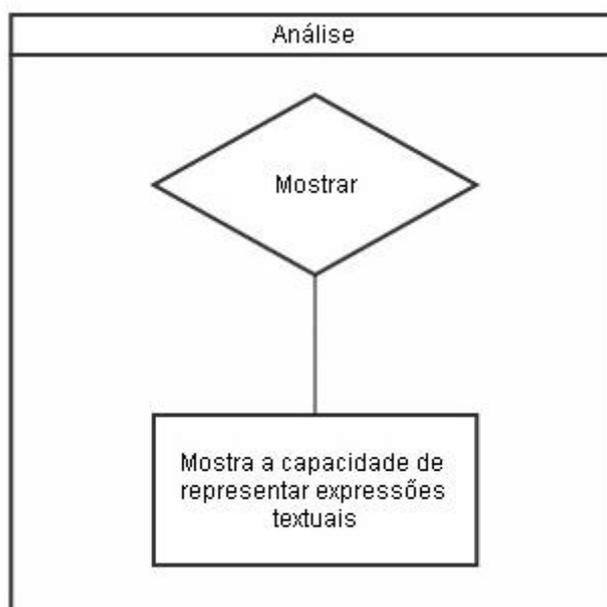


Figura 8 - Tema na Fase de Análise

Fonte: Adaptado de Clarke; Baniassad (2005, p. 55)

Nesta figura um tema é representado por um losango, o retângulo representa a descrição de uma funcionalidade particular dos requisitos e a ligação entre estes representa o relacionamento. Neste caso, o tema *Mostrar* representa uma característica do sistema e dentro do retângulo à descrição de um requisito relevante. Há um relacionamento entre o tema *Mostrar* e o requisito, por este apresentar em sua descrição uma atividade referente ao tema *Mostrar*.

O relacionamento entre temas pode ser representado de duas formas: *Concept Sharing* ou *Crosscutting*. No relacionamento *Concept Sharing* diferentes temas apresentam elementos que representam o mesmo núcleo de conceitos dentro de um domínio. Esses elementos podem ser classes, métodos ou atributos e o conceito relata algo sobre o domínio.

O relacionamento por *Crosscutting* ocorre quando temas podem “atravessar” uns aos outros. Um tema é transversal a outro se seu comportamento é desencadeado pelo comportamento de outro.

Em temas com relacionamento por *Concept Sharing* os temas relacionados podem operar sem o conhecimento do outro, diferente de temas que apresentam relacionamento por *Crosscutting*, os quais dependem de um conhecimento abstrato dos temas que estão atravessando não lhes permitindo operar independentemente.

Para a aplicação deste método, primeiramente deve ser feita a análise dos requisitos do sistema examinando a documentação do sistema de requisitos.

Esta fase é denominada *Finding Themes* (Buscando Temas) e possui quatro principais atividades: *Escolha dos conjuntos de temas em potencial*, *Refinamento do conjunto de temas*, *Identificação de quais temas são aspectos* e *Elaboração para a fase de projeto*. A Figura 9 ilustra as atividades desenvolvidas pelo método.

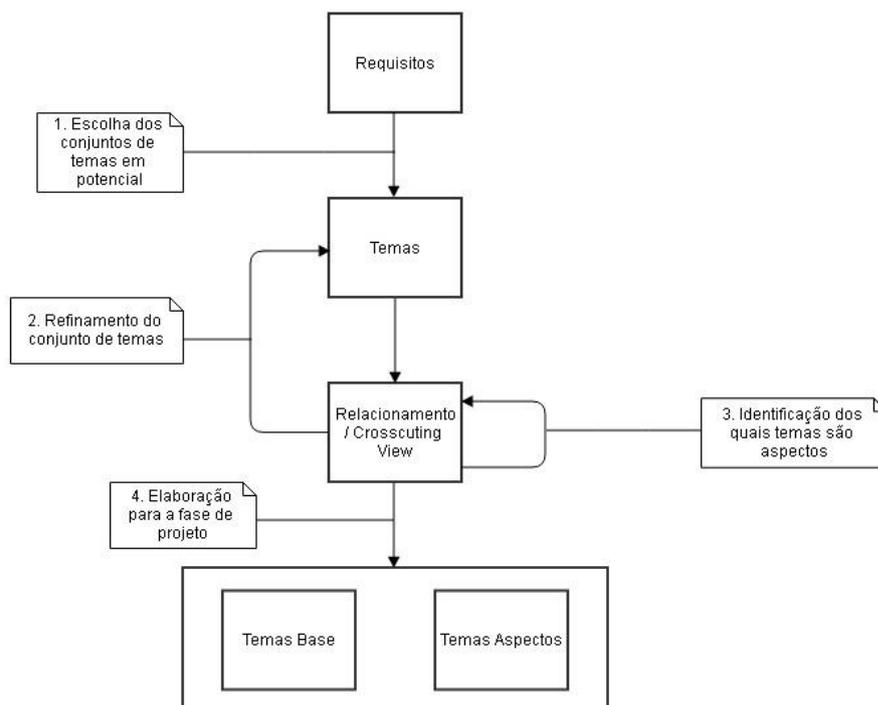


Figura 9 - Etapas do Método Baseado em Temas

Fonte: Adaptado de Clarke; Baniassad (2005, p. 63)

A primeira atividade, *Escolha dos Conjuntos de Temas em Potencial*, tem como finalidade identificar o conjunto de preocupações dos requisitos, então, da mesma forma em que se examinam os requisitos buscando por entidades-chave, é feita a busca por preocupações-chave. Em seguida se realiza uma iteração sobre o conjunto, decidindo se deve adicionar, retirar, dividir ou agrupar temas. A proposta desta primeira atividade é escolher temas em potencial, diminuindo uma eventual lista de características ou objetos para o projeto.

A atividade de *Refinamento do Conjunto de Temas* permite identificar quais dos potenciais temas são grandes o suficiente para serem modelados separadamente. Para esta etapa de refinamento são utilizados alguns operadores, tais como: adicionar, excluir, dividir e agrupar.

Para o refinamento de temas é utilizado o relacionamento por vista (*Relationship view*), o qual recebe como entrada o conjunto de requisitos e de temas. A Figura 10 ilustra um exemplo de relacionamento por vista.

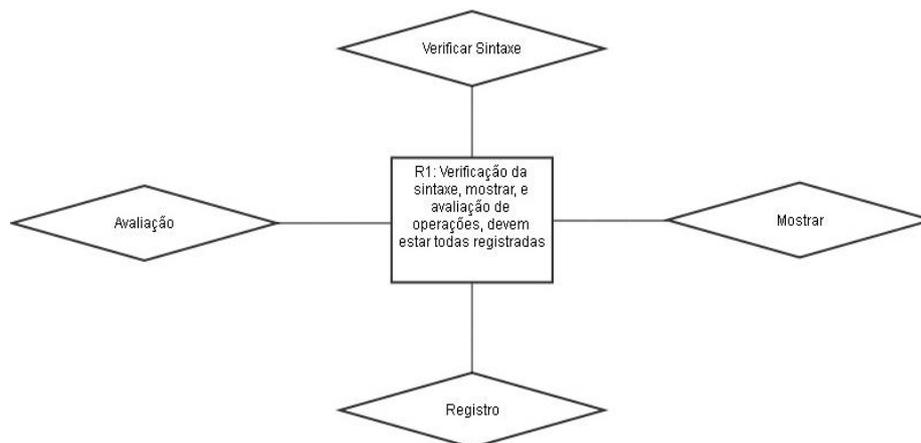


Figura 10 - Exemplo de Relacionamento por vista
Fonte: Adaptado de Clarke; Baniassad (2005, p. 65)

Conforme ilustrado na Figura 10, os requisitos são representados por um retângulo e os temas por um losango. Caso o nome de um tema seja mencionado em um requisito, então há um relacionamento por vista entre estes.

Este estágio tem por objetivo observar qual o papel representado pelo comportamento nos requisitos, a fim de determinar se estes são temas ou apenas comportamentos (por exemplo, métodos) dentro de temas. Este processo de decisão sobre potenciais temas ou não é feito de forma iterativa e intuitiva. É possível que no próximo passo de identificação de temas transversais, seja feito um novo refinamento dos temas escolhidos.

A próxima atividade de *Identificação de Temas Transversais* é utilizada à vista de relacionamento entre temas obtido no passo anterior, em que a identificação de aspectos é feita observando os requisitos compartilhados entre os temas. Requisitos compartilhados representam o entrelaçamento de responsabilidades, o que pode ser identificado como um aspecto no sistema.

Conforme ilustrado na Figura 10 o requisito R1 é compartilhado com diversos temas, sendo um candidato a aspecto. Porém, para identificá-lo como um aspecto este deve obedecer a três regras: Divisão não funciona (não é possível reescrever, de forma a isolar ou remover o compartilhamento), dominância quer dizer associação (caso o requisito não possa ser dividido, isto quer dizer que este deve estar associado a um tema dominante. A associação de um requisito com um tema dominante denota que este é um aspecto), e desencadeamento de aspectos base (aspectos são desencadeados por

comportamentos base. Se dois temas são mencionados em um requisito, o tema o qual é desencadeado pelo outro é classificado como um aspecto, e o que desencadeia é o tema do tipo base). O tema que se identifica com as regras anteriores, pode ser descrito como um aspecto. Esta figura ilustra o relacionamento do tipo transversal (*Crosscutting*) do aspecto identificado com os temas que partilham do mesmo requisito.

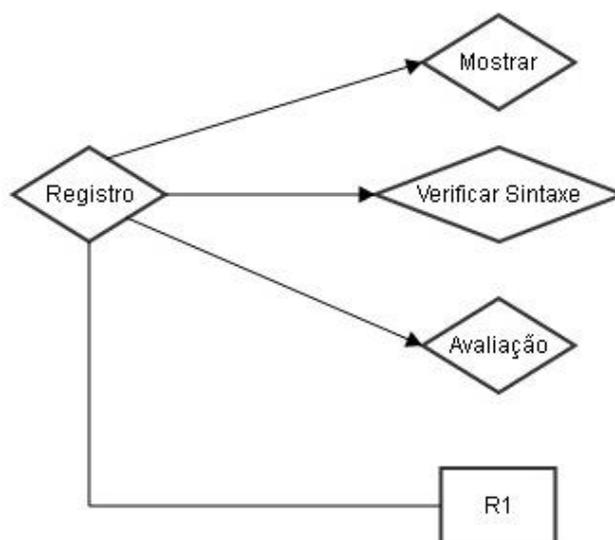


Figura 11 - Relacionamento Transversal entre Temas
 Fonte: Adaptado de Clarke; Baniassad (2005, p. 67)

O tema Registro obedece as três regras descritas anteriormente, sendo considerado um aspecto. Este tema compartilha o requisito R1 com outros três temas (Mostrar, Verificar Sintaxe e Avaliação), portanto, Registro possui um relacionamento Transversal com os outros três temas.

Por fim, a atividade de *Preparação para a fase de projeto*, consiste em separar a partir do relacionamento por vista, quais temas são considerados Temas Base e quais são Temas Aspectos, utilizando de *Vista de Temas Base* (*Viewing Base Themes*) e *Vista de Temas Aspecto* (*Viewing Aspect Themes*).

A vista de temas base é a vista individual de temas os quais não atravessam (*crosscutting*) outros temas. Cada tema é observado de uma vista particular, os quais são mostrados como requisitos associados ao tema, ações derivadas dos requisitos e entidades escolhidas inicialmente. A Figura 12 ilustra um exemplo da *Vista de Temas Base*.

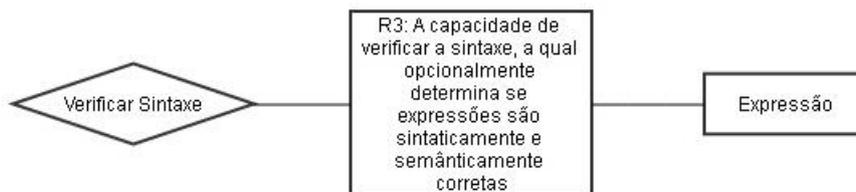


Figura 12 - Exemplo de Vista de Temas Base

Fonte: Adaptado de Clarke; Baniassad (2005, p. 68)

O relacionamento do tema *Verificar Sintaxe* com o requisito R3 e o retângulo menor ligado ao requisito R3 representa uma entidade, no caso, uma expressão.

Através desta vista é possível prever algumas classes e métodos. Temas e outras funcionalidades descritas poderão ser traduzidas em métodos e entidades em classes.

A *Vista de Temas Aspecto* é um pouco diferente da *de Temas Base*, pois nesta vista são tratados elementos que foram atravessados pelo tema observado, e não há a presença de elementos descritos nos requisitos os quais são internos a este tema. A Figura 13 ilustra um exemplo de *Vista de Temas Aspecto*.

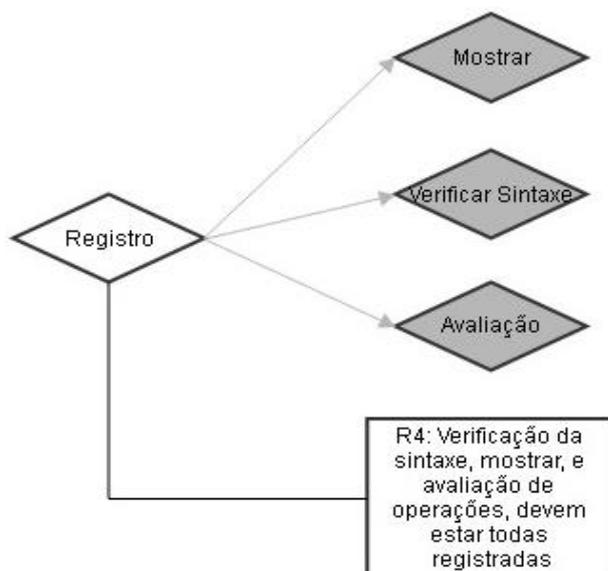


Figura 13 - Exemplo de Vista de Temas Aspecto

Fonte: Adaptado de Clarke; Baniassad (2005, p. 70)

Neste exemplo um *Registro* é o tema com relacionamento do tipo *Crosscutting*, os outros temas (Mostrar, Verificar Sintaxe e Avaliação) ligados ao *Registro* são os elementos atravessados, e o Requisito R4, representa o requisito que entrelaça estes temas. Este tipo de vista tem por objetivo descrever como temas com relacionamento *Crosscutting* afetam outros temas no sistema.

2.2.3 Identificação de Candidatos a Aspectos Baseado em Requisitos

O método proposto por Rashid et al. (2005) é um modelo genérico para a Orientação a Aspectos na Engenharia de Requisitos (*Aspect Oriented Requirements Engineering*), o qual é capaz de separar relacionamentos do tipo *Crosscutting* por meio de requisitos funcionais e não-funcionais.

Este modelo para identificação de preocupações transversais (*Crosscutting Concerns*) é composto de seis etapas ilustradas na Figura 14.

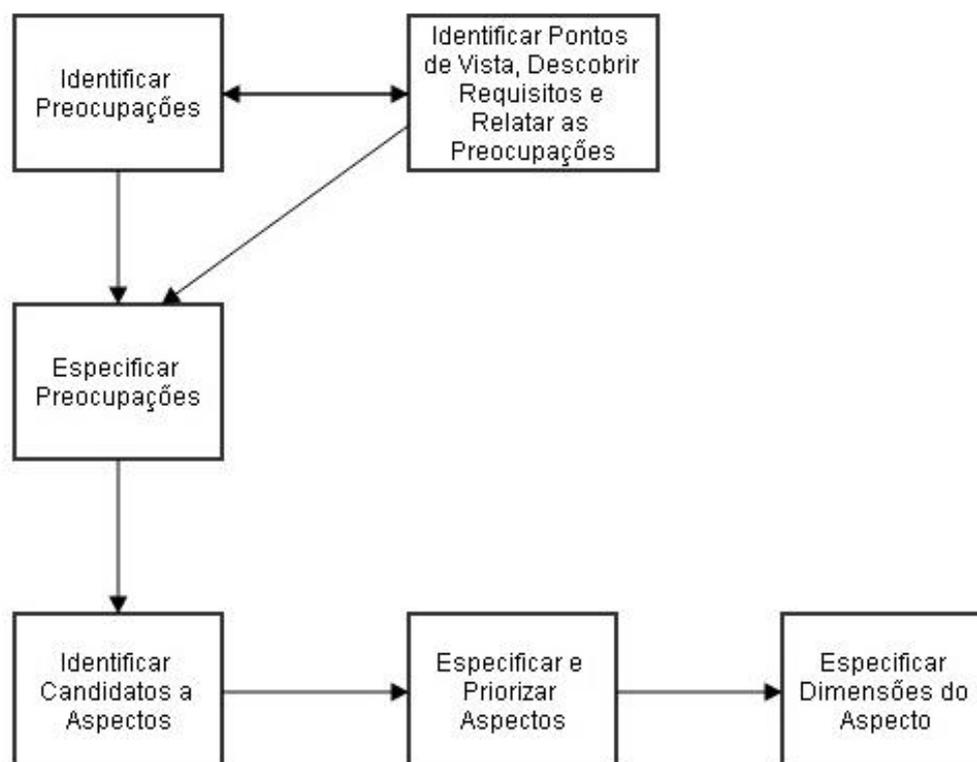


Figura 14 - Modelo de Identificação de Candidatos a Aspectos a partir de Requisitos

Fonte: Adaptado de Rashid et al. (2005)

Primeiramente é necessário fazer a identificação de preocupações e requisitos. Esta etapa pode ocorrer de forma paralela, pois segundo Rashid et al. (2005): "A ordem na qual estas duas atividades são realizadas é dependente da dinâmica de interação entre engenheiros de requisitos e os *Stakeholders*".

A identificação de Preocupações é feita analisando os requisitos iniciais encontrados. Nesta primeira etapa também são identificados os pontos de vista (*View Points*) obtidos por meio do documento de requisitos. Pontos de vista podem ser definidos como um pedaço de cada preocupação que é relevante para o *Stakeholder* (RASHID et al., 2005).

A próxima etapa *Especificar Preocupações* é executada depois de selecionadas as preocupações do passo anterior. A Figura 15 ilustra um exemplo de especificação de preocupações.

Preocupação: Tempo de Resposta.
Requisitos Externos:

1. O portão do pedágio tem de reagir em tempo, seguindo a seguinte ordem:
 - 1.1 Ler o dispositivo identificador;
 - 1.2 Acender a luz (para verde ou amarelo) antes do motorista deixar a área do portão do pedágio
 - 1.3 Mostrar a quantia a ser paga, antes do motorista deixar a área do portão do pedágio
 - 1.4 Fotografar o número da placa da retaguarda de veículos não autorizados
2. O sistema necessita reagir em tempo quando:
 - 2.1 O usuário ativar o dispositivo utilizando um ATM

Figura 15 - Exemplo de Especificação de Preocupações: Tempo de Resposta

Fonte: Adaptado de Rashid et al. (2005)

A preocupação identificada nesta figura é o *Tempo de Resposta* sendo esta relacionada aos requisitos numerados em 1 e 2.

Depois de identificadas e especificadas preocupações e requisitos, é feito o relacionamento de preocupações, pontos de vista e requisitos utilizando um *template* denominado *JPREView*. A Figura 16 ilustra um exemplo deste relacionamento aplicado em dois pontos de vista.

<p><i>Ponto de Vista:</i> ATM.</p> <p><i>Preocupações:</i> Segurança, Compatibilidade, Tempo de Resposta.</p> <p><i>Requisitos:</i></p> <ol style="list-style-type: none"> 1. O ATM irá encaminhar o número de cartão do cliente, número da conta, e o dispositivo (<i>gizmo</i>) identificador para o sistema de ativação. 2. O ATM irá enviar o número da conta para o sistema para obter o dispositivo identificador associado com a conta. 3. O ATM irá enviar o número da conta, novo número de cartão, e o dispositivo identificador para o sistema para atualizar o número do cartão e reativar o dispositivo.
<p><i>Ponto de Vista:</i> Ferramenta de Saída.</p> <p><i>Preocupações:</i> Tempo de Resposta, Correção, Questões Legais.</p> <p><i>Requisitos:</i></p> <ol style="list-style-type: none"> 1. O motorista irá ver uma luz amarela caso ele não use uma entrada para o pedágio. 2. A quantia a ser debitada, depende do ponto de entrada.

Figura 16 - Exemplo de Pontos de Vista de Preocupações em JPREView

Fonte: Adaptado de Rashid et al. (2005)

Especificadas as preocupações utilizando a forma *JPREView*, o analista poderá iniciar a identificação de candidatos a aspectos, em que se uma preocupação atravessa (*Crosscutting*) vários requisitos, então esta passa a ser um candidato a aspecto.

Na Figura 15 os requisitos 1 e 2 da preocupação *Tempo de Resposta* atravessam os requisitos de dois diferentes pontos de vista, *ATM* e *Ferramenta de Saída* ilustrados na Figura 16. Consequentemente a preocupação *Tempo de Resposta* pode ser qualificada como um candidato a aspecto.

A etapa de *Especificação e Priorizar Aspectos* fornece através da especificação, uma oportunidade de refinamento dos aspectos a fim de torná-los mais concretos. A priorização permite descrever o grau que um aspecto pode restringir um ponto de vista.

Por fim, a etapa de *Especificar Dimensões do Aspecto* especifica a dimensão de um candidato a aspecto tornando possível determinar se o aspecto identificado irá ter influência em estágios mais avançados de desenvolvimento e identificar se o candidato a aspecto poderá ser uma função, decisão ou um aspecto. O Quadro 1 ilustra o mapeamento de candidatos a aspectos e suas influências.

Candidato a Aspecto	Influência	Mapeamento
Compatibilidade	Especificação, Arquitetura, Projeto, Evolução	Função
Tempo de Resposta	Arquitetura, Projeto	Aspecto
Questões Legais	Especificação	Função
Segurança	Arquitetura, Projeto	Aspecto

Quadro 1 - Mapeamento de Candidatos a Aspectos e suas Influências

Fonte: Adaptado de Rashid et al. (2005)

Neste quadro observa-se que o candidato a aspecto *Compatibilidade* apresenta requisitos que influenciam a Especificação, Arquitetura, Projeto e Evolução. Entretanto, o aspecto *Compatibilidade* será mapeado como uma função, por permitir a ativação e reativação de dispositivos.

Por outro lado, o candidato a aspecto *Tempo de Resposta* influencia os requisitos referentes à arquitetura e ao projeto e será mapeado como aspecto na fase de projeto e de implementação, pois suas propriedades não podem ser encapsuladas em uma única classe, ou seja, estão espalhadas por meio delas.

2.2.4 Identificação de Aspectos Baseado em Atributos de Requisitos Não-Funcionais

O método proposto por Siqueira da Silva (2012) se baseia na identificação de aspectos a partir de similaridades com Requisitos Não-Funcionais (RNF). Os Requisitos Funcionais tem por objetivo representar o que o cliente deseja, enquanto que os RNFs estão relacionados a padrões de qualidade, tais como: segurança, confiabilidade, desempenho (XAVIER, 2009).

A Figura 17 ilustra o processo deste método por meio de um diagrama de atividades da *Unified Modeling Language* (UML).

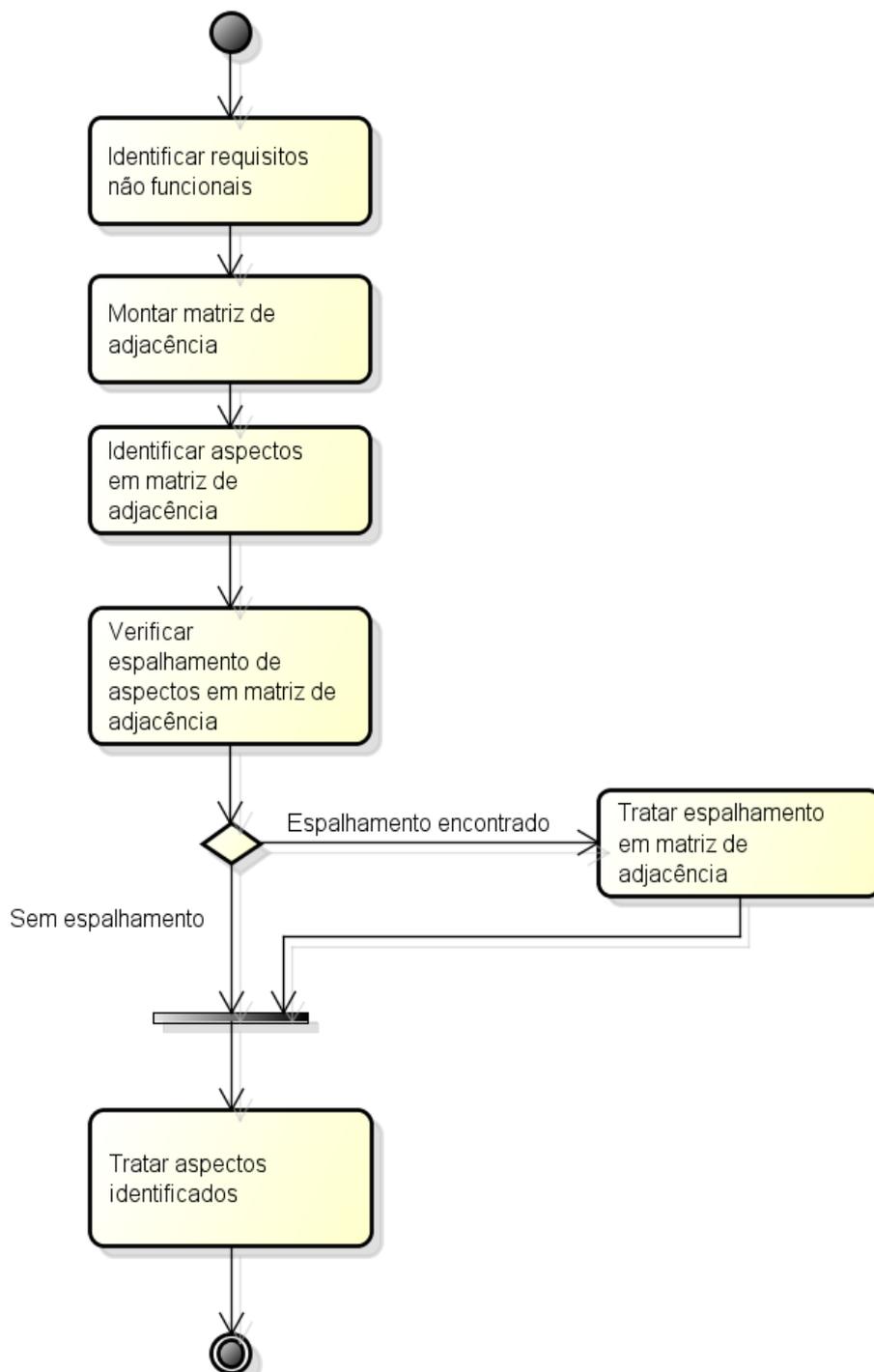


Figura 17 - Processo do Método de Siqueira da Silva (2012)

Fonte: Siqueira da Silva (2012, p. 37)

A primeira atividade *Identificar Requisitos Não-Funcionais* é responsável pela identificação dos RNFs. Os requisitos não-funcionais são divididos em atributos conforme proposto por Mairizia et al. (2010), a qual identificou cinco RNFs mais utilizados no desenvolvimento de software: desempenho, confiabilidade, segurança, usabilidade e manutenibilidade. Cada

um destes RNFs é composto por vários atributos. O Quadro 2 ilustra um exemplo de RNFs com seus atributos.

Desempenho	
D1	Tempo de Resposta
D2	Capacidade
D3	Latência
D4	<i>Throughput</i>
D5	Processamento
D6	Velocidade de Execução
D7	<i>Transit Delay</i>
D8	Carga de Trabalho
D9	Utilização de Recursos
D10	Uso de Memória
D11	Precisão
D12	Eficiência
D13	<i>Delay</i>
D14	Perda de Dados
D15	Processamento de Transação Concorrente
D16	<i>Garbage Collector</i>

Quadro 2 - Requisito Não-Funcional Desempenho e seus respectivos atributos

Fonte: Siqueira da Silva (2012, p. 40)

Identificados os RNFs e seus atributos, a próxima atividade é montar a matriz de adjacência. Esta matriz é criada por meio dos RNFs e das descrições textuais de caso de uso do sistema. O Quadro 3 ilustra um modelo dessa matriz.

Título de Matriz					
	A1	A2	A3	A...	An
TF1P1					
TF...P...					
TFmPt					

Quadro 3 - Modelo Matriz de Adjacência

Fonte: Siqueira da Silva (2012, p. 42)

O elemento *Título da Matriz* representa o nome do caso de uso, a variável *P* se refere a um passo do caso de uso, *TF* representa o tipo de fluxo,

podendo este representar *Fluxo Básico* (FB), Fluxo Alternativo (FA) e Subfluxo (S), e o elemento *A* representa um atributo de RNF.

Depois de montada a matriz de adjacência é realizada a atividade de *Identificar Aspectos em Matriz de Adjacência*, que é responsável por identificar os aspectos e seus possíveis espalhamentos.

O analista deve verificar os RNFs e seus atributos dentro dos passos da descrição textual. Caso um atributo seja identificado em algum dos passos, este atributo deve ser marcado, sinalizando que há dentro daquele passo um aspecto. Este processo é repetido para todos os RNFs levantados. O Quadro 4 ilustra o resultado deste processo considerando os atributos referentes ao RNF de desempenho.

Realizar Cadastro de Obra								
	D ₁	D...	D ₅	D ₆	D...	D ₁₁	D...	D ₁₅
FB P ₁								
FB P ₂	X			X				
FB P ₃				X				
FB P ₄				X		X		
FB P ₅	X		X	X		X		
FB P ₆								X

Quadro 4 - Matriz de adjacência para a descrição textual Realizar Cadastro de Obra

Fonte: Siqueira da Silva (2012, p. 43)

Depois de preenchidas todas as matrizes de adjacência para todos os casos de uso, analisando os atributos de cada RNF, a próxima atividade é *Verificar Espalhamento de Aspectos em Matriz de Adjacência e Tratar Espalhamento*, pois é possível visualizar o espalhamento de aspectos pelas matrizes. A Figura 18 ilustra um exemplo de espalhamento, em que nas matrizes de adjacência *Consulta de Cadastro de Obra* e *Realizar Cadastro de Obra* os aspectos espalhados são D1, D6 e D15. O espalhamento encontrado será tratado com o uso de uma matriz de adjacência auxiliar.

Consulta de Cadastro de Obra								
	D ₁	D...	D ₅	D ₆	D...	D ₁₁	D...	D ₁₅
FB P ₁								
FB P ₂								
FB P ₃	X			X				X

Cadastro de Obra								
	D ₁	D...	D ₅	D ₆	D...	D ₁₁	D...	D ₁₅
FB P ₁								
FB P ₂	X			X				
FB P ₃				X				
FB P ₄				X		X		
FB P ₅	X		X	X		X		
FB P ₆								X

Figura 18 - Espalhamento de Aspectos em Matrizes Adjacência

Fonte: Siqueira da Silva (2012, p. 45)

O Quadro 5 ilustra a matriz auxiliar em que o elemento *Título da Matriz Auxiliar* indica o nome da matriz, a variável *UC* representa a identificação do caso de uso, *P* se refere a um passo da descrição textual, *TF* representa o tipo de fluxo, podendo esse ser do tipo *FB* (Fluxo Básico ou Principal), *FA* (Fluxo Alternativo) ou *S* (Subfluxo), e a variável *A* se refere aos atributos identificados na primeira atividade, onde *n* é o total de atributos para cada tipo de RNF. O número de matrizes auxiliares será proporcional ao número de RNFs.

Título da Matriz Auxiliar				
	A ₁	A ₂	A...	A _n
UC ₁ - TF ₁ P ₁				
UC... - TF... P...				
UC _x - TF _m P _t				

Quadro 5 - Matriz auxiliar utilizada para tratar espalhamento de aspectos

Fonte: Siqueira da Silva (2012, p. 46)

O Quadro 6 ilustra um exemplo de preenchimento da matriz auxiliar, com base no espalhamento detectado na Figura 18 para os aspectos do tipo desempenho. Caso nenhum espalhamento seja identificado, a matriz auxiliar não existirá.

Matriz auxiliar – Aspectos do Tipo Desempenho								
	D ₁	D...	D ₅	D ₆	D...	D ₁₁	D...	D ₁₅
Cadastro de Obra – FB P ₂	X			X				
Cadastro de Obra – FB P ₃				X				
Cadastro de Obra – FB P ₄				X		X		
Cadastro de Obra – FB P ₅	X		X	X		X		
Cadastro de Obra – FB P ₆								X
Consultar Cadastro de Obra – FB P ₃	X			X				X

Quadro 6 - Matriz auxiliar preenchida com aspectos do tipo desempenho

Fonte: Siqueira da Silva (2012, p. 48)

Por fim, é realizada a última atividade, *Tratar os Aspectos Identificados*, onde a entrada são as matrizes originais e as auxiliares. Desta forma, é possível visualizar a representação dos aspectos dentro das descrições textuais de um caso de uso.

O Quadro 7 ilustra um modelo de descrição textual utilizado para representar os aspectos identificados, o qual é composto por *Identificador do Caso de Uso* – código que identifica o caso de uso –, *Título da Descrição Textual* – representa o nome do caso de uso –, *Breve Descritivo* – refere-se a algumas informações sobre o caso de uso –, *Tipo de Fluxo* – indica o tipo de fluxo: se é do tipo principal (FB), alternativo (FA) ou subfluxo (S) –, *P* – designa os passos dos casos de uso –, *Descrição do Passo* – corresponde a descrição do passo –, *Join Point* – retrata a ocorrência de aspecto dentro do respectivo passo – e *Tipo de Aspecto* – representa os tipos de aspectos os quais foram identificados nessa descrição textual–.

<<Identificador do caso de uso>>	<<Título da descrição textual>>
<< Breve descritivo >>	
<<Tipo de fluxo>>	
TF ₁ P ₁	<< descrição do passo >> <<Join point >>
TF... P...	<< descrição do passo >> <<Join point >>
TF _m P _t	<< descrição do passo >> <<Join point >>
Tipos de aspectos	
<<Tipo de aspecto ₁ >>	
<<Tipo de aspecto... >>	
<<Tipo de aspecto _j >>	

Quadro 7 - Descrição Textual Utilizada para Representar os Aspectos Identificados

Fonte: Siqueira da Silva (2012, p. 49)

Além de estruturar os aspectos com as descrições de caso de uso, o método proposto se utiliza do conceito de *Pointcut* para representar o tipo de aspecto identificado no caso de uso. No Quadro 8 o *Pointcut* é composto dos elementos: *UC* que representa qual o caso de uso, *P* representa os passos do fluxo e *TF* representa os tipos de fluxo.

<<Tipo do aspecto >>	
UC ₁ – TF ₁ P ₁	<<nome do atributo>>
UC... – TF... P...	<<nome do atributo>>
UC _a – TF _m P _t	<<nome do atributo>>

Quadro 8 - *Pointcut* Representando um Tipo de Aspecto

Fonte: Siqueira da Silva (2012, p. 50)

Os conceitos de *Join Point* e *Pointcut* são semelhantes ao utilizado na Programação Orientada a Aspectos (POA), onde um *Join Point* é um ponto específico da execução de um programa, tal como: a chamada de um método ou a construção de um objeto, sendo que estes pontos permitem a alteração de seu comportamento por meio de aspectos. *Pointcut* é a declaração do ponto em que se deseja interromper a execução do programa para que seja inserido outro comportamento.

Neste método um *Join Point* se refere a um passo da descrição textual, enquanto que um *Pointcut* diz respeito a um tipo de aspecto. Após a realização dessa atividade, os aspectos identificados estão preparados para serem utilizados nas próximas atividades do ciclo de desenvolvimento de software.

2.3 ANÁLISE DOS MÉTODOS DE IDENTIFICAÇÃO DE ASPECTOS NA FASE DE ANÁLISE

Esta seção apresenta uma breve análise dos métodos, exibidos no Quadro 9, citados na seção 2.2, relatando suas principais características e dificuldades segundo alguns autores tais como: Kossoski (2010), Siqueira da Silva (2012), Sampaio et al. (2005a) e Bakker et al. (2005).

Método	Características	Dificuldades
Identificação de Aspectos Baseada em <i>Use Case Slice</i> (JACOBSON; NG, 2004).	Uso de <i>Use Case Slice</i> como forma de encapsular aspectos durante a fase de análise. Propõe uma representação para aspectos da <i>Unified Modeling Language</i> (UML) (UML, 2012).	Não é capaz de identificar todos os aspectos do sistema (KOSSOSKI, 2010). Exige que os pessoais responsáveis pela análise tenham conhecimento sobre aspectos (SIQUEIRA DA SILVA, 2012).
Identificação de Aspectos Baseada em Temas (CLARKE; BANASSAD, 2005)	Uso de temas para o tratamento de aspectos.	O analista deve analisar todo o documento de requisitos, selecionando palavras às quais podem ser compreendidas como temas (SAMPAIO et al. 2005a).
Identificação de Candidatos a Aspectos Baseado em Requisitos (RADHID et al., 2005)	Separação de relacionamentos <i>crosscutting</i> com base em requisitos funcionais e não-funcionais. Requisitos com preocupações transversais podem ser considerados candidatos a aspectos.	Não possui suporte para as outras fases do ciclo de vida de desenvolvimento de software (BAKKER et al., 2005).
Identificação de Aspectos Baseado em Atributos de Requisitos Não-Funcionais (SIQUEIRA DA SILVA, 2012)	Identificação de aspectos a partir de similaridades com requisitos não-funcionais.	Requer grande atenção do analista para a montagem e preenchimento da matriz de adjacência e na identificação e avaliação do espalhamento de aspectos.

Quadro 9 – Análise dos Métodos
Fonte: Autoria Própria

Com a análise das características e dificuldades dos métodos estudados, observa-se que cada um deles possui uma maneira diferenciada para identificação de aspectos. Neste trabalho optou-se por estudar mais aprofundamente o de Siqueira da Silva (2012) por: ser baseado em requisitos não-funcionais, não possuir uma ferramenta automatizada, contemplar a

identificação dos aspectos por meio dos cenários dos casos de uso e ser possível verificar o espalhamento de aspectos entre os casos de uso.

2.4 FERRAMENTAS DE IDENTIFICAÇÃO DE ASPECTOS NA FASE DE ANÁLISE OU FERRAMENTAS SIMILARES

Nesta seção serão descritas ferramentas encontradas na literatura que tem a capacidade de identificar aspectos durante a fase de análise.

2.4.1 Theme/Doc

Theme/Doc utiliza a abordagem em temas, a qual tem por objetivo a identificação de temas no documento de requisitos, utilizando os conceitos e passos descritos na seção 2.2.2 provendo vistas e suporte de funcionalidades para a identificação e representação de aspectos para a fase de análise.

Utilizando as vistas Theme/Doc o desenvolvedor é capaz de visualizar o relacionamento entre os comportamentos descritos no documento de requisitos, e assim determinar quais dos comportamentos podem ser do tipo base e quais são do tipo *Crosscutting*.

O primeiro passo antes de utilizar a ferramenta é determinar o conjunto inicial de temas, em que nessa primeira etapa, temas podem ser apenas um conjunto de preocupações as quais foram julgadas pelo analista como importantes. Esse conjunto de temas poderá ser refinado de forma iterativa pelo analista depois de geradas as vistas de ação (*action view*), pois a partir destas vistas, é possível observar se há um conjunto de temas os quais façam sentido.

Depois de gerado o primeiro conjunto de temas pode-se dar início a geração das primeiras vistas de ação.

Uma vista de ação na ferramenta Theme/Doc é criada utilizando como entrada, a lista de ações-chave (temas) identificadas pelo desenvolvedor através da análise do documento de requisitos separando verbos sensíveis, e os requisitos descritos no documento de requisitos original. Depois de

informado os dados, a ferramenta realiza uma análise léxica do texto, gerando as vistas de ação.

Na Figura 19 é ilustrada uma vista de ação criada pela ferramenta Theme/Doc, em que os losangos representam as ações identificadas, e os requisitos são representados pelos retângulos arredondados. Caso um requisito apresente uma referência para com uma ação chave, então há um relacionamento entre a ação e o requisito.

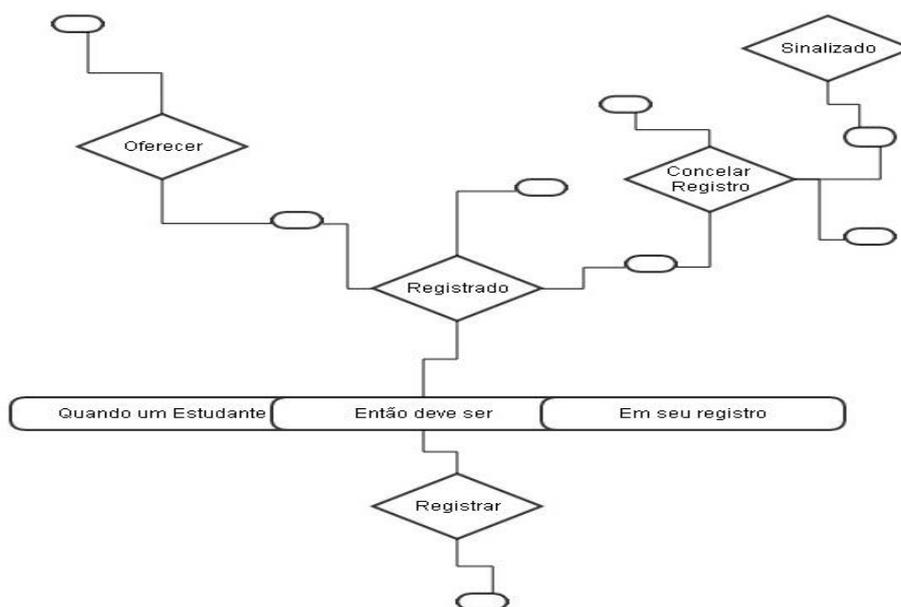


Figura 19 - Vista de Ação dos Requisitos do CMS

Fonte: Adaptado de Baniassad; Clarke (2004)

Através desta vista é possível visualizar que há entrelaçamento de comportamentos com os requisitos, como por exemplo, *Registrado*, mencionado em quatro requisitos.

Caso os requisitos se refiram a mais de um comportamento, neste pode haver o relacionamento do tipo *Crosscutting*. A identificação de aspectos é feita com base nestes requisitos compartilhados, separando as ações em dois grupos: base e *crosscutting*.

Para realizar essa separação é utilizada a funcionalidade *Clipping*, a qual examina os requisitos compartilhados, verificando qual ação está mais acoplada a qual requisito. Esta funcionalidade utiliza como validação de requisitos compartilhados as três regras descritas em 2.2.2, *Dominância não*

Funciona, Dominância quer dizer Associação e Desencadeamento de Aspectos Base.

Após separadas as ações em base e *Crosscutting*, a ferramenta gera a vista de ação *clipped*.

A Figura 20 ilustra um exemplo de vista de ação *clipped*, em que os temas do tipo *crosscutting* se encontram acima dos outros temas, os quais ele atravessa, e a flecha cinza indica o *crosscutting* na vista *clipped*.

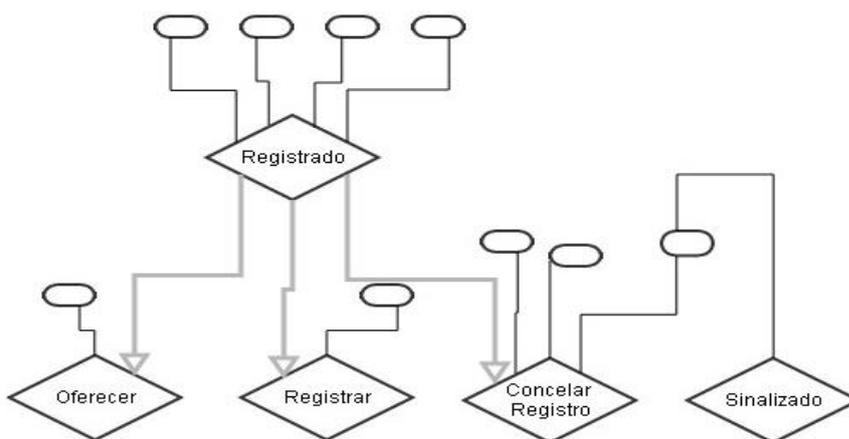


Figura 20 - Vista de Ação Clipped
Fonte: Adaptado de Baniassad; Clarke (2004)

A partir da vista de ação *Clipped*, cada agrupamento se torna um tema, o qual será modelado na fase de projeto através da ferramenta Theme/UML.

2.4.2 Minerando Aspectos em Requisitos

A ferramenta de mineração de aspectos em requisitos descrita por Sampaio et al. (2005a), provê uma ferramenta capaz de identificar aspectos durante a fase de análise, com o auxílio de um processador de linguagem natural, denominada WMATRIX.

Esta ferramenta é baseada no método descrito na seção 2.2.3, em que algumas das etapas ocorrem de forma semi-automática.

Primeiramente o analista entra com o arquivo contendo a descrição textual dos requisitos solicitados pelo cliente no processador de linguagem natural WMATRIX, o qual a partir de um critério pré-estabelecido pelo analista,

tal como: substantivos ou verbos, realiza o processamento das descrições textuais.

Os primeiros elementos identificados pela ferramenta, conforme os passos descritos na Figura 14, são os pontos de vista, em que os critérios para esta identificação são em geral substantivos. O analista poderá visualizar por meio da ferramenta a frequência com que palavras específicas ocorrem. Esta abordagem permite ao analista rapidamente minerar pontos de vista e seus requisitos relacionados.

Os próximos elementos identificados são os requisitos transversais, os quais comumente são verbos de ação. Da mesma forma como ocorre com os pontos de vista, o analista tem a possibilidade de visualizar a frequência com que os verbos de ação ocorrem, e ainda visualiza os trechos do texto que estão estes verbos, facilitando a identificação de relacionamentos do tipo transversal, e de candidatos a aspecto.

Outra forma de identificar requisitos transversais segundo o autor é mapeando requisitos não-funcionais, porém estes conceitos estão implícitos no texto e são complexos de serem minerados automaticamente, sendo melhores julgados por analistas.

Depois de levantados os pontos de vista e os requisitos transversais são realizados os passos *Especificar Preocupações*, *Identificar Candidatos a Aspectos*, *Priorizar e Especificar Aspectos* e *Especificar Dimensões do Aspecto*, conforme descrito em 2.2.3.

Esta ferramenta propõe uma abordagem parcialmente automatizada de mineração de aspectos através da identificação de preocupações e mapeamento de palavras de ação, gerando maior agilidade para a execução dos passos seguintes do método.

2.5 ANÁLISE DAS FERRAMENTAS DE IDENTIFICAÇÃO DE ASPECTOS NA FASE DE ANÁLISE

Nesta seção é feita uma análise das ferramentas apresentadas anteriormente, relatando suas características, vantagens e dificuldades. Esta análise é exibida no Quadro 10.

Ferramenta	Características	Vantagens	Dificuldades
Theme/Doc (BANIASSAD; CLARKE, 2004)	Provê vistas e suporte de funcionalidades para a identificação e representação de aspectos.	Geração de vistas de ação e vista de ação <i>clipped</i> de forma automatizada facilitando a identificação de temas <i>crosscutting</i> e base.	O analista deve analisar todo o documento separando palavras do tipo chave para formar o conjunto de temas. O conjunto de temas necessita ser refinado de forma iterativa e intuitiva.
Minerando Aspectos em Requisitos (SAMPAIO et al., 2005a)	Identificação de candidatos a aspecto através do processamento de descrições textuais de requisitos solicitados pelo cliente.	Identificação de pontos de vista e requisitos transversais de forma semi-automatizada	RNFs nem sempre são detectados.

Quadro 10– Análise das Ferramentas
Fonte: Autoria Própria

Com base na análise das características, verificou-se que é viável o desenvolvimento de uma ferramenta capaz de identificar aspectos com base no método proposto por Siqueira da Silva (2012). Apesar da ferramenta de Minerando Aspectos usar requisitos não funcionais, a mesma nem sempre os detecta como sendo um aspecto, sendo que o foco é analisar o documento de requisitos.

3 ASPECT-NFR: UMA FERRAMENTA PARA DETECÇÃO DE ASPECTOS

Este capítulo apresenta a ferramenta semi-automatizada proposta para identificação de aspectos na fase de análise. A Seção 3.1 relata o processo de como o método implementado funciona. A Seção 3.2 apresenta a modelagem da ferramenta. Por fim, a Seção 3.3 descreve um exemplo do uso da ferramenta.

3.1 PROCESSO DE IMPLEMENTAÇÃO DO MÉTODO DE SIQUEIRA DA SILVA

O método implementado contém duas atividades: Informar Dados de Entrada (processo manual) e Identificar Candidatos a Aspectos (processo automatizado). Cada atividade possui um conjunto de etapas e a identificação de candidatos a aspectos ocorre quando todas forem cumpridas. É importante ressaltar que o método proposto por Siqueira da Silva (2012) não contempla a identificação de aspectos espalhados pelos diversos sistemas e subsistemas, porém o processo proposto contempla esta funcionalidade. O processo de implementação do método está ilustrado na Figura 21.

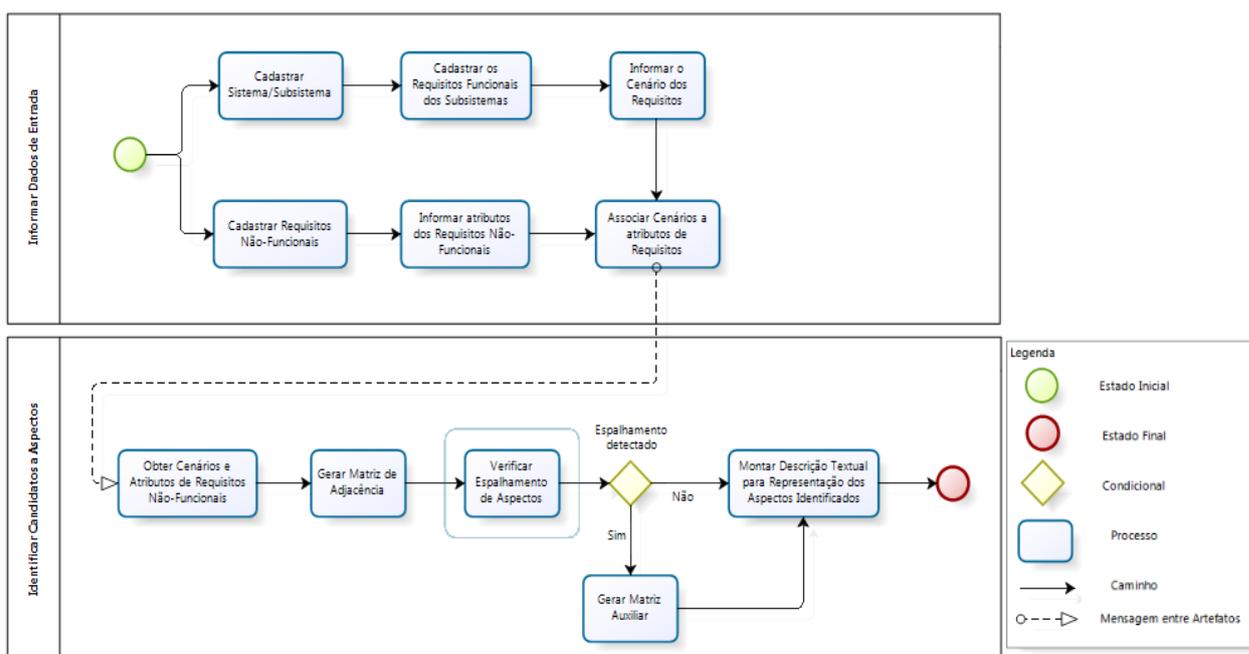


Figura 21 – Processo do Método.

Fonte: Aatoria Própria

A primeira atividade, Informar Dados de *Entrada*, contempla o conjunto de etapas de cadastro de informações relevantes para a aplicação do método. O processo se inicia com o *Cadastrar Sistema/Subsistema* ou com o *Cadastrar Requisitos Não-Funcionais*, ou seja, essas etapas podem ocorrer simultaneamente.

O processo *Cadastrar Sistema/Subsistema* é responsável pelo cadastro do sistema a ser analisado e os subsistemas referentes a ele, como por exemplo, pode-se ter um sistema capaz de calcular o preço de venda de um produto/serviço e este contempla os seguintes subsistemas: *Login*, *Cálculo* e *Atributos*. Já o processo *Cadastrar Requisitos Não-Funcionais* é referente ao cadastro dos RNFs, tais como: desempenho, segurança e robustez.

Depois de realizados estes cadastros iniciais, o analista poderá prosseguir para a etapa de *Cadastrar os Requisitos Funcionais dos Subsistemas*, caso ele tenha realizado o cadastro do sistema/subsistema ou para a etapa de *Informar atributos dos Requisitos Não-Funcionais*, caso tenha efetuado o processo de cadastro dos RNFs. O *Cadastrar os Requisitos Funcionais dos Subsistemas* se refere à entrada de dados de descrições textuais as quais foram solicitadas para o desenvolvimento do sistema, como por exemplo, o requisito funcional de *Logar no Sistema*.

A etapa *Informar atributos dos Requisitos Não-Funcionais* é efetuado após o cadastro dos RNFs, pois este se refere à associação dos atributos de RNFs com seus respectivos RNFs, por exemplo, o RNF desempenho apresenta como atributos: tempo de resposta, capacidade, latência, processamento e velocidade de execução.

Cadastrados os Requisitos Funcionais dos subsistemas, o analista deverá informar seus respectivos cenários. Este cenário contém um nome, uma breve descrição sobre sua finalidade, os passos percorridos por este caso de uso que podem ser: básico, alternativo ou subfluxo. A Figura 22 ilustra um exemplo de descrição textual para o requisito funcional de Logar no Sistema.

Descrição do Caso de Uso: Logar no Sistema
Descrição breve Este caso de uso inicia quando o usuário executa a aplicação.
Fluxo Básico B1. Logar no Sistema O caso de uso inicia quando um usuário quer acessar o sistema. <ol style="list-style-type: none"> 1. O sistema pede que o usuário informe usuário e senha. 2. O usuário digita o seu nome de usuário e sua senha no sistema e clica no botão logar. 3. O sistema verifica a inserção dos dados através de <u>Conferir Login</u>. 4. O sistema autoriza o acesso do usuário, mostrando-lhe a tela inicial do sistema com suas configurações pessoais. 5. O caso de uso termina.
Fluxo Alternativo A1. Preenchimento de Campos No passo 3 caso ocorra algum erro no preenchimento de campos obrigatórios ou na validação dos dados informados, o sistema limpará os campos e mostrará a tela de <i>login</i> novamente.

Figura 22 - Exemplo de Descrição Textual Logar no Sistema

Fonte: Siqueira da Silva (2012, p. 55)

Cadastradas estas informações, deverá ser feita a etapa de *Associar Cenários a atributos de Requisitos*, a qual consiste em associar os passos com os atributos de um determinado RNF. Esta associação é feita para cada RNF cadastrado previamente. O Quadro 11 ilustra um exemplo de associação de atributos do RNF Desempenho com a descrição textual *Logar no Sistema*.

Logar no Sistema – Desempenho																
	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	D ₈	D ₉	D ₁₀	D ₁₁	D ₁₂	D ₁₃	D ₁₄	D ₁₅	D ₁₆
FB P ₁																
FB P ₂	X					X										
FB P ₃	X					X			X	X					X	X
FB P ₄	X					X										
FB P ₅																
FA ₁ P ₁	X					X			X	X					X	X

Quadro 11– Atributos Desempenho para a Descrição Logar no Sistema

Fonte: Siqueira da Silva (2012, p. 81)

O título da matriz é composto do caso de uso e o RNF analisados, as variáveis *FB* e *FA* indicam o tipo do fluxo, sendo *FB* fluxo básico e *FA* fluxo alternativo, a variável *P* ilustra os passos do fluxo, e *D* representa os atributos do RNF *Desempenho* conforme ilustrado pelo Quadro 2. O símbolo *X* ilustra os pontos dos passos em que ocorre o atributo de RNF.

Finalizada a associação, dá-se início as atividades de *Identificar Candidatos a Aspectos*. Primeiramente há a etapa *Obter Cenários e Atributos de Requisitos Não-Funcionais* em que se utiliza os dados cadastrados na primeira atividade.

Em seguida, a etapa *Gerar Matriz de Adjacência* consiste em montar uma matriz de adjacência contendo os passos de cada cenário associados aos atributos de um determinado RNF. Este processo é realizado para todos os cenários e para cada RNF. O Quadro 11 ilustrou um exemplo de matriz de adjacência gerado para o RNF *Desempenho* associado à descrição textual *Logar no Sistema*.

Geradas todas as matrizes de adjacência, a etapa *Verificar Espalhamento de Aspectos* consiste em comparar as matrizes geradas, cruzando as informações cadastradas dos passos e atributos de forma a verificar o espalhamento de aspectos, podendo esta etapa ser realizada para a análise entre sistemas ou entre subsistemas. A Figura 23 ilustra um exemplo de espalhamento de aspectos.

Logar no Sistema – Desempenho																
	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16
FB P1																
FB P2	X					X										
FB P3	X					X			X	X					X	X
FB P4	X					X										
FB P5																
FA1 P1	X					X			X	X					X	X

Conferir Login – Desempenho																
	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16
FB P1	X				X					X						
FB P2						X				X	X	X				
FB P3									X							
FA1 P1																

Figura 23 – Exemplo de Espalhamento de Aspectos

Fonte: Adaptado de Siqueira da Silva (2012)

Neste exemplo o espalhamento de aspectos é indicado pelo símbolo X que ilustra os pontos em que ocorrem os atributos do RNF *Desempenho* e os retângulos em vermelho ilustram o espalhamento de aspectos entre os casos de uso. Caso haja espalhamento de aspectos, efetua-se a etapa *Gerar Matriz Auxiliar* a qual gera uma nova matriz contendo os atributos do RNF avaliado, os cenários e seus respectivos passos. A Figura 24 ilustra um exemplo de matriz auxiliar para o RNF *Desempenho*.

Matriz Auxiliar – Desempenho																
	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16
Conferir Login – FB P1	X									X						
Conferir Login – FB P2						X				X						
Conferir Login – FB P3									X							
Logar no Sistema – FB P2	X					X										
Logar no Sistema – FB P3	X					X			X	X						
Logar no Sistema – FB P4	X					X										
Logar no Sistema – FA1 P1	X					X			X	X						

Figura 24 – Exemplo de Matriz Auxiliar para o RNF Desempenho

Fonte: Siqueira da Silva (2012, p. 86)

Esta figura ilustra um exemplo de matriz auxiliar para o RNF *Desempenho*, em que a variável *D* ilustra os atributos para RNF *Desempenho*, *Logar no Sistema* e *Conferir Login* são os casos de uso avaliados, *FB* representa fluxo Básico, *FA* ilustra fluxo alternativo e *P* representa o passo do caso de uso e o símbolo *X* representa os pontos em que ocorrem atributos de RNF.

Geradas as matrizes, a etapa *Montar Descrição Textual para Representação dos Aspectos Identificados* consiste do preenchimento de uma descrição textual contendo: Código de identificação, título do caso de uso, breve descrição do caso de uso, os passos do caso de uso com seus referentes fluxos e descrições. Caso o passo tenha a ocorrência de um aspecto este deverá apresentar o estereótipo <<*Join Point*>>. Ao final da descrição deverão ser listados os aspectos identificados no caso de uso. O Quadro 12 ilustra um exemplo de aspectos dentro da descrição textual *Logar no Sistema*.

UC001	Logar no Sistema
Breve Descritivo:	Este Caso de Uso inicia quando o usuário executa a aplicação.
Fluxo Principal	
P1- O sistema pede que o usuário informe usuário e senha. Join point	
P2- O usuário digita o seu nome de usuário e sua senha no sistema e clica no botão logar. Join point	
P3- O sistema verifica a inserção dos dados através de <u>Preencher Campos Obrigatórios</u> . Join point	
P4- O sistema autoriza o acesso do usuário, mostrando-lhe a tela inicial do sistema com suas configurações pessoais. Join point	
P5- O caso de uso termina.	
Fluxo Alternativo	
FA1 – No passo 3 caso ocorra algum erro no preenchimento de campos obrigatórios ou na validação dos dados informados, o sistema limpará os campos e mostrará a tela de <i>login</i> novamente. Join point	
Tipos de aspectos	
Segurança	
Usabilidade	
Manutenibilidade	
Confiabilidade	
Desempenho	

Quadro 12– Exemplo de aspectos dentro da descrição textual *Logar no Sistema*
Fonte: Siqueira da Silva (2012, p. 59)

Além de estruturar os aspectos com as descrições de caso de uso, é gerado um *Pointcut* que ilustra textualmente um determinado aspecto identificado, contendo o caso de uso com seu receptivo passo de ocorrência do aspecto e a qual atributo está relacionado o passo. O Quadro 13 ilustra um exemplo de *Pointcut* para o aspecto *Desempenho*.

Aspecto Desempenho	
Conferir Login – FB P ₁	D ₁ , D ₅ , D ₁₀ .
Conferir Login – FB P ₂	D ₆ , D ₁₀ , D ₁₁ , D ₁₂ .
Conferir Login – FB P ₃	D ₉ .
Logar no Sistema – FB P ₂	D ₁ , D ₆ .
Logar no Sistema – FB P ₃	D ₁ , D ₆ , D ₉ , D ₁₀ , D ₁₅ , D ₁₆ .
Logar no Sistema – FB P ₄	D ₁ , D ₆ .
Logar no Sistema – FA ₁ P ₁	D ₁ , D ₆ , D ₉ , D ₁₀ , D ₁₅ , D ₁₆ .

Quadro 13– Exemplo de *Pointcut* para o Aspecto Usabilidade
Fonte: Siqueira da Silva (2012)

Nota-se neste quadro que *Logar no Sistema* e *Conferir Login* representam os casos de uso os quais apresentam aspectos do tipo *Desempenho*, *FB* ilustra o fluxo do tipo básico, *FA* representa fluxo do tipo alternativo, *P* representa o passo do caso de uso, e *D* ilustra o atributo para o RNF *Desempenho*.

3.2 MODELAGEM DA FERRAMENTA

A modelagem lógica e de banco para a ferramenta proposta foram criadas usando o diagrama de classes UML e o Modelo Entidade Relacionamento, respectivamente. A Figura 25 ilustra o diagrama de classes utilizado para a modelagem lógica do método.

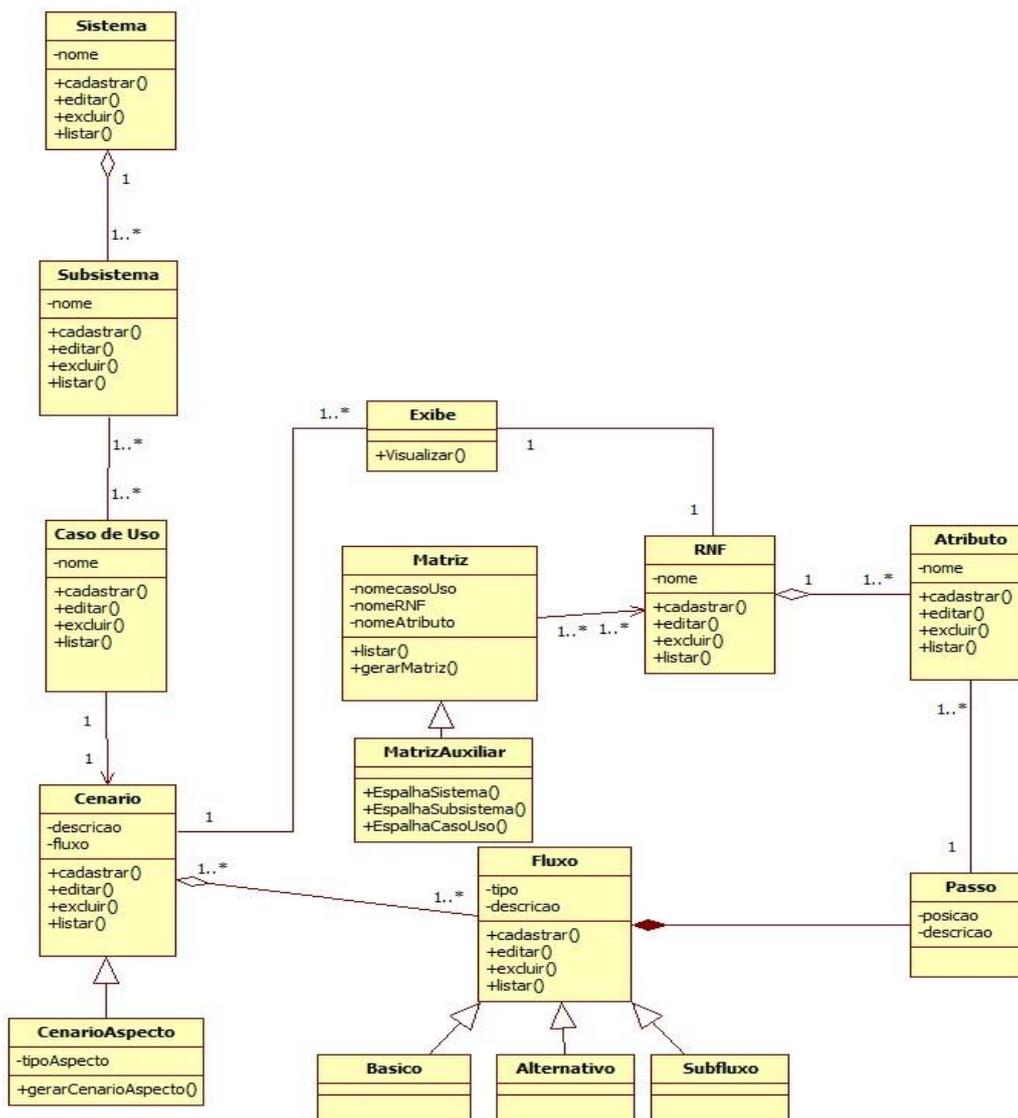


Figura 25 - Diagrama de Classes da Ferramenta AspectNFR

Fonte: Autoria Própria

Foram modeladas as classes *Sistema*, *Subsistema*, *Caso de uso*, *RNF* e *Atributo* apresentando como atributo um *nome* e como métodos: *cadastrar()*, *editar()*, *excluir()* e *listar()*.

A superclasse *Cenario* é composta pelos atributos: *descrição* e *fluxo*, e pelos métodos *cadastrar()*, *editar()*, *excluir()* e *listar()*. A subclasse *CenarioComAspecto*, além dos atributos e métodos herdados da superclasse *Cenario*, apresenta ainda o atributo *tipoAspecto* e o método *gerarCenarioAspecto()*.

Um *Cenario* é composto de fluxos, desta forma foi necessária a modelagem da classe *Fluxo* a qual apresenta como atributos *tipo* (básico,

alternativo ou subfluxo) e *descrição*, e como métodos *cadastrar()*, *editar()*, *excluir()* e *listar()*.

Cada fluxo de um cenário é composto por passos, em que estes possuem como atributos *posição* e *descrição*, e como método *vincular()*.

A superclasse *Matriz* é composta pelos atributos *nomeCasoUso*, *nomeRNF* e *nomeAtributo*, e como métodos *listar()* e *gerarMatriz()*. O método *listar()* é responsável por exibir na página inicial os casos de uso os quais podem ser visualizados como matriz e o método *gerarMatriz()* tem a finalidade de gerar a matriz com os atributos de um RNF e os passos do casos de uso escolhido. Sua subclasse *MatrizAuxiliar*, apresenta os métodos *EspalhaSistema()*, *EspalhaSubsistema()* e *EspalhaCasoUso()* os quais são responsáveis por gerar a matriz auxiliar de espalhamento de aspectos entre sistemas no caso do método *EspalhaSistema()*, entre subsistemas através do método *EspalhaSubsistema()* e entre casos de uso utilizando o método *EspalhaCasoUso()*.

Por fim, foi criada a classe *Exibe* que foi desenvolvida para a exibição dos tipos de aspectos identificados, os quais estão relacionados a casos de uso conforme ilustrado no Quadro 13.

A Figura 26 ilustra a modelagem do banco de dados do método por meio do Modelo Entidade Relacionamento.

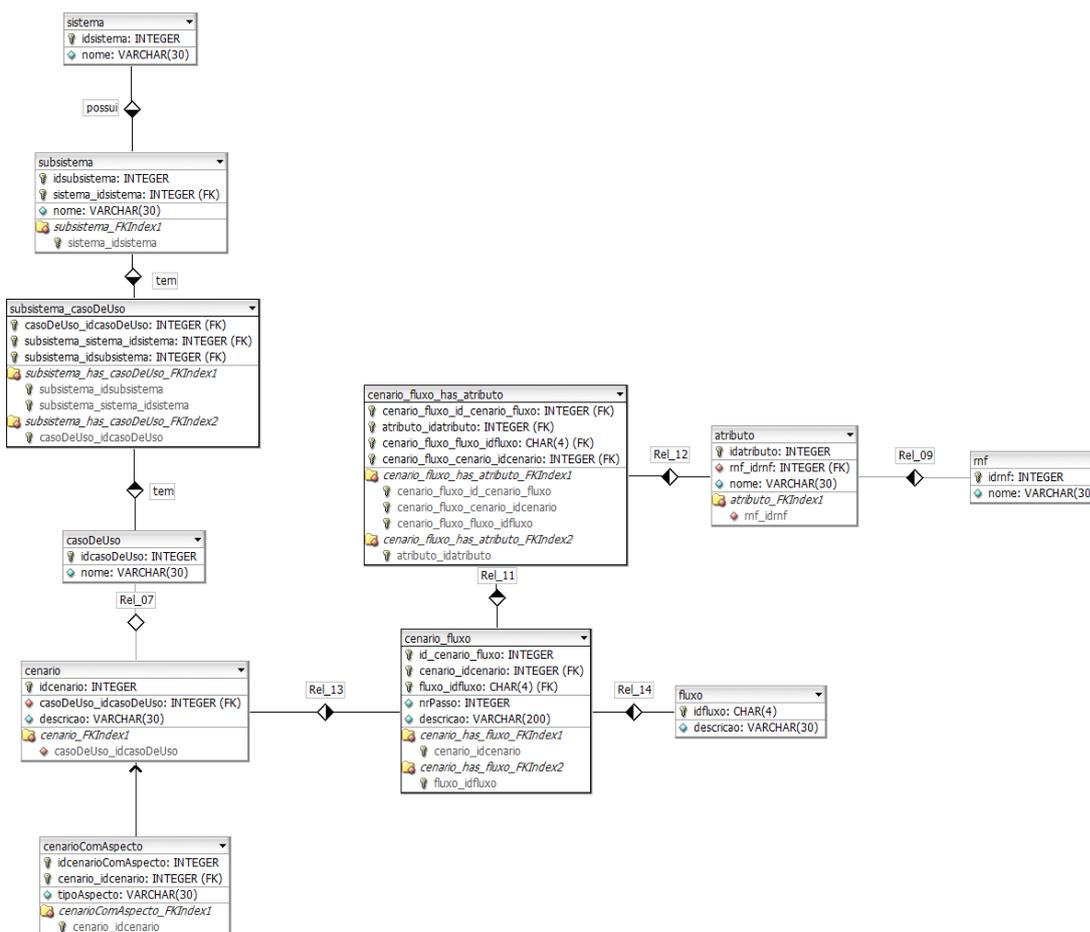


Figura 26 – Modelo Entidade Relacionamento da Ferramenta AspectNFR

Fonte: Autoria própria

Este modelo é similar ao modelo lógico desenvolvido, a não ser pelo fato de que neste não há a tabela para a geração da matriz, sendo esta gerada apenas no momento de execução da ferramenta. A classe *Passo* não foi modelada como uma tabela por se tratar de um campo enumerado, sendo este agregado à tabela *cenario_fluxo*, tabela esta resultante do relacionamento entre as tabelas *cenario* e *fluxo*.

A implementação da ferramenta foi feita usando a linguagem PHP e JavaScript, as telas foram desenvolvidas utilizando HTML e o banco de dados usado foi o MYSQL. Trechos de códigos não serão exibidos neste trabalho, pois o objetivo é mostrar o processo de implementação e sua modelagem.

3.3 UM EXEMPLO DO USO DA FERRAMENTA PROPOSTA

Esta seção apresenta um exemplo prático do uso da ferramenta proposta. Primeiramente se desenvolveu uma tela inicial que permite o acesso aos cadastros, edições, exclusões e aos elementos que compõe o método implementado. A Figura 27 ilustra a tela inicial desenvolvida.

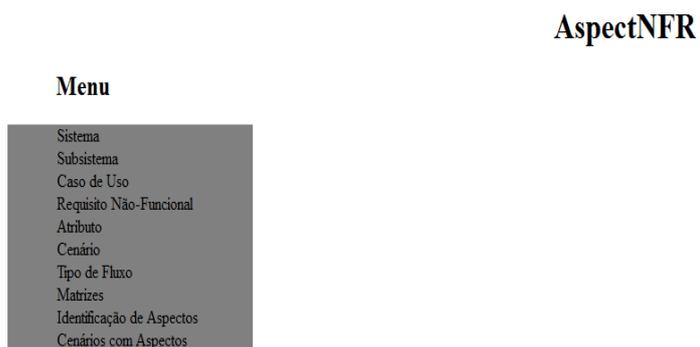


Figura 27 – Página Principal da Ferramenta AspectRNF

Fonte: Autoria Própria

O primeiro passo do método é o cadastro de Sistema, Requisito Não-Funcional ou Tipo de Fluxo. Acionando o *link* presente na aba Sistema será exibida a página de listagem de sistemas. Caso não haja sistemas cadastrados, a página ficará como ilustrado na Figura 28 em que se exibe a mensagem “Não há Sistemas Cadastrados” e um botão de “Novo Sistema”.

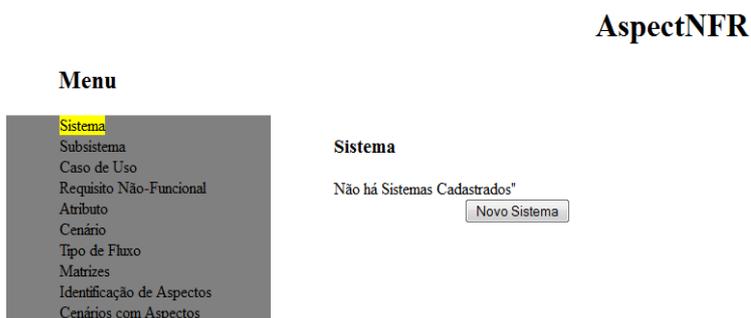


Figura 28 – Página Sistema

Fonte: Autoria Própria

Ao clicar neste botão, o analista é redirecionado para a tela de cadastro de sistema conforme ilustrado pela Figura 29.

Sistema

Nome do Sistema

Gravar

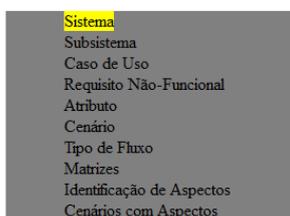
Figura 29 – Página de Cadastro de Sistema

Fonte: Autoria Própria

Um sistema tem como atributo seu nome. Como exemplo realizou o cadastro do sistema de nome “Framemk”. Este sistema está sendo desenvolvido pelo Grupo de Pesquisa em Sistemas de Informação, Linha Engenharia de Software, do Câmpus Ponta Grossa. A Figura 30 ilustra como ficará a tela de listagem de sistemas cadastrados.

AspectNFR

Menu



Sistema

Nome do Sistema		
Framemk	Editar	Excluir
Novo Sistema		

Figura 30 – Página de Listagem de Sistema

Fonte: Autoria Própria

Todos os passos descritos para a inclusão de um Sistema se repetem quando do cadastro de um RNF.

Para o Tipo de Fluxo o cadastro permite gravar um nome para o fluxo com no máximo duas letras e uma breve descrição sobre o tipo de fluxo conforme ilustra a Figura 31.

Tipo de Fluxo

Nome do Tipo (Máximo duas letras)

Descrição

Gravar

Figura 31 – Página de Cadastro de Tipo de Fluxo

Fonte: Aatoria Própria

A funcionalidade de Tipo de Fluxo permite o cadastro de outros tipos de fluxo além dos três apresentados por Siqueira da Silva (2012): Alternativo, Básico e Sub-Fluxo.

Depois de confirmado o cadastro de um Sistema ou RNF, o analista poderá realizar o cadastro de um Subsistema ou Atributo.

Caso não haja sistemas ou atributos cadastrados, o método não permite que sejam cadastrados subsistemas ou atributos conforme ilustrado na Figura 32.

AspectNFR

<p>Menu</p> <ul style="list-style-type: none"> Sistema Subsistema Caso de Uso Requisito Não-Funcional Atributo Cenário Tipo de Fluxo Matrizes Identificação de Aspectos Cenários com Aspectos 	<p>Subsistema</p> <p>Não há Sistemas cadastrados!</p>
--	--

Figura 32 – Página de Validação de um Subsistema

Fonte: Aatoria Própria

Porém, caso haja sistemas pré-cadastrados o sistema irá exibir uma tela similar a do Sistema conforme ilustrado na Figura 33.

AspectNFR

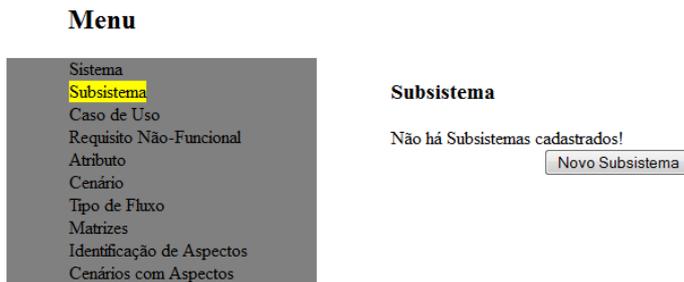


Figura 33 – Página Subsistema

Fonte: Aatoria Própria

Ao clicar sobre o botão “Novo Subsistema” é feito um redirecionamento para uma tela de cadastro de subsistemas conforme ilustra a Figura 34.

Subsistema

The image shows a registration form for a subsystem. It has a text input field for 'Nome do Subsistema' containing the text 'Controle Login'. Below it is a dropdown menu for 'Sistema' with 'Framemk' selected. At the bottom of the form is a button labeled 'Gravar'.

Figura 34 – Página de Cadastro de Subsistema

Fonte: Aatoria Própria

Esta figura mostra o cadastro do subsistema “Controle Login”, o qual foi vinculado ao sistema “Framemk”.

A tela de cadastro de atributo é um pouco diferente da de subsistema conforme ilustra a Figura 35.

Atributo

Requisito Não-Funcional
Desempenho ▾
Nome do Atributo

Alias

Figura 35 – Página de Cadastro de Atributo

Fonte: Autoria Própria

O cadastro de atributo apresenta o RNF ao qual está vinculado, seu nome e um apelido (alias) para o preenchimento da matriz, como por exemplo D1 para o atributo “Tempo de Resposta” pertencente ao RNF Desempenho.

Finalizados os cadastros de sistema, subsistema, RNF e atributo de RNF, poderá ser feito o cadastro de caso de uso. A Figura 36 ilustra a tela de cadastro de caso de uso.

Caso de Uso

Nome do Caso de Uso

Sistema
Framemk ▾
Subsistema

Figura 36 – Página de Cadastro Caso de Uso

Fonte: Autoria Própria

Um caso de uso deverá ser vinculado a um Sistema e a um Subsistema, e deverá apresentar um nome. Como exemplo será utilizado o caso de uso “Logar no Sistema”.

Após cadastrado o caso de uso, na listagem de caso de uso do menu principal irá aparecer o caso de uso vinculado ao sistema e subsistema ao qual este pertence conforme apresenta a Figura 37.

AspectNFR

Menu

- Sistema
- Subsistema
- Caso de Uso
- Requisito Não-Funcional
- Atributo
- Cenário
- Tipo de Fluxo
- Matrizes
- Identificação de Aspectos
- Cenários com Aspectos

Caso de Uso

Nome do Caso de Uso	Nome do Subsistema	Nome do Sistema		
Logar no Sistema	Framemk	Controle Login	Editar	Excluir

Figura 37 – Página de Listagem de Caso de Uso

Fonte: Autoria Própria

O próximo passo é realizar o cadastro de um Cenário. A Figura 38 ilustra o cadastro desenvolvido para cenário.

Cadastro de Cenário

Sistema

Subsistema

Nome do Caso de Uso

Breve Descritivo:

FB - Fluxo básico

FA - Fluxo Alternativo

S - Sub Fluxo

Figura 38 – Página de Cadastro de Cenário

Fonte: Autoria Própria

O cadastro de um cenário apresenta alguns atributos tais como: sistema, subsistema e caso de uso a ser vinculados, breve descrição, fluxos e os passos de cada fluxo.

Primeiramente deve ser selecionado o sistema ao qual se deseja vincular o cenário. Ao selecionar um sistema, a ferramenta irá buscar em sua base de dados os subsistemas vinculados ao sistema selecionado, listando-os no *combobox* Subsistema. O mesmo processo da seleção do subsistema irá gerar o preenchimento do *combobox* dos casos de uso vinculados ao subsistema selecionado.

Em seguida, será preenchido o campo *Breve Descritivo* o qual deve conter uma breve descrição do cenário. Prosseguindo deverão ser preenchidos os passos para seu correspondente tipo de fluxo. Os fluxos os quais estão aparecendo no cadastro de cenário são os fluxos previamente cadastrados.

Ao clicar no botão *Adicionar* abaixo de um tipo de fluxo será criado um novo passo conforme ilustra a Figura 39.

Cadastro de Cenário

Sistema
Framemk ▾

Subsistema
Controle Login ▾

Nome do Caso de Uso
Logar no Sistema ▾

Breve Descritivo:

FB - Fluxo básico

P1

FA - Fluxo Alternativo

S - Sub Fluxo

Este caso de uso inicia quando o usuário executa a aplicação

Figura 39 - Página de Cadastro Cenário com Passo

Fonte: Autoria Própria

Ao clicar sobre o botão *Adicionar* abaixo do *Tipo de Fluxo FB*, foi gerado um passo *P1* o qual apresenta um campo para adicionar a descrição do passo e dois botões: *Excluir* e *Vincular*. O botão *Excluir* permite a exclusão do passo

do cenário e o botão *Vincular* irá permitir a vinculação de atributos de RNFs ao passo selecionado. A Figura 40 ilustra como é realizada esta associação.

Vincular Atributos de Requisitos Não-Funcionais

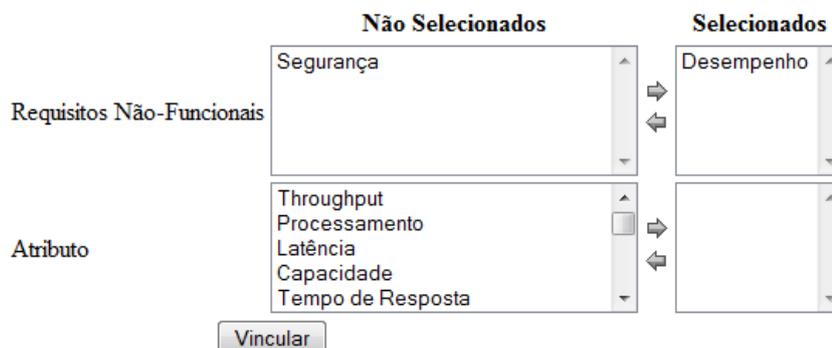


Figura 40 – Página para Vincular atributos de RNFs ao Passo

Fonte: Autorial Própria

Nesta figura são listados os RNFs previamente cadastrados e ao selecionar um destes, será preenchido os campos atributos os quais poderão ser vinculados com o passo. Nesta etapa poderão ser vinculados quantos atributos forem necessários e de diferentes RNFs, como por exemplo, podem ser selecionados os RNFs *Segurança* e *Desempenho*, listando assim seus atributos no campo atributos. Depois de listados todos os atributos disponíveis, os selecionados poderão ser vinculados ao passo. Depois de concluída esta associação o sistema retorna ao cenário.

Após concluídos o preenchimento dos passos e a vinculação de atributos ao passo, o cenário poderá ser gravado e é possível a visualização da matriz de adjacência. A Figura 41 ilustra um exemplo da pagina inicial.

AspectNFR

Menu

Sistema
Subsistema
Caso de Uso
Requisito Não-Funcional
Atributo
Cenário
Tipo de Fluxo
Matrizes
Identificação de Aspectos
Cenários com Aspectos

Matriz

Nome do Caso de Uso	Descrição do Cenário	
Logar no Sistema	Este caso de uso inicia quando o usuário executa a aplicação.	Desempenho <input type="button" value="Visualizar"/>

Figura 41 – Página de Listagem da Matriz por Caso de Uso

Fonte: Autoria Própria

Nessa figura foram vinculados atributos do RNF *Desempenho* para o caso de uso “Logar no Sistema”. Abaixo do RNF há o botão *Visualizar*, ao clicá-lo a matriz será gerada, conforme apresenta a Figura 42.

Matriz

Logar no Sistema - Desempenho

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16
FB1																
FB2	X					X										
FB3	X					X			X	X					X	X
FB4	X					X										
FB5																
FA1	X					X			X	X					X	X

Figura 42 – Página de Visualização da Matriz de Adjacência “Logar no Sistema” – RNF – Desempenho

Fonte: Autoria Própria

As colunas D1 a D16 representam os atributos de RNF conforme ilustrados no Quadro 2, os passos FB1 a FB5 ilustram os fluxos básicos, FA1 ilustra o fluxo alternativo cadastrados para o cenário. As células marcadas com um “X” em vermelho ilustram os pontos em que há o vínculo entre atributo do RNF e o passo do caso de uso.

Outra matriz que pode ser visualizada é a auxiliar que ilustra o espalhamento de aspectos entre os casos de uso. A Figura 43 ilustra a tela com a página *Espalhamento de Aspectos*.



Figura 43 – Página Inicial do Espalhamento de Aspectos

Fonte: Aatoria Própria

Conforme ilustrado na Figura 43, a ferramenta possibilita a visualização do espalhamento entre sistemas, subsistemas e caso de uso. A Figura 44 ilustra a tela após selecionar uma das opções de visualização.



Figura 44 – Página Inicial do Espalhamento de Aspectos Selecionando a Opção Caso de Uso

Fonte: Aatoria Própria

Ao selecionar a opção *Caso de Uso*, a ferramenta exibe os casos de uso cadastrados. Após selecionar os casos de uso desejados, é necessário escolher o RNF. Selecionado o RNF será possível verificar o espalhamento. A Figura 45 ilustra um exemplo do espalhamento de aspectos sendo selecionados os casos de uso *Conferir Login* e *Logar no Sistema* e o RNF *Desempenho*.

Matriz Auxiliar

Requisito Não-Funcional: Desempenho

Caso de Uso: Conferir Login - Logar no Sistema

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16
Conferir Login - FB1	X									X						
Conferir Login - FB2						X				X						
Conferir Login - FB3									X							
Logar no Sistema - FA1	X					X			X	X					X	X
Logar no Sistema - FB2	X					X										
Logar no Sistema - FB3	X					X			X	X					X	X
Logar no Sistema - FB4	X					X										

Figura 45 - Visualização Espalhamento de Aspectos por meio da Matriz Auxiliar

Fonte: Autoria Própria

A ferramenta gerou a matriz de espalhamento de aspectos contendo os passos dos casos de uso selecionados, com os atributos de RNF do RNF *Desempenho*. As células marcadas com um “X” em vermelho ilustram os pontos em que há o vínculo entre atributo do RNF e o passo do caso de uso.

Finalizadas as visualizações de matrizes, outra opção da ferramenta é a visualização de cenários com aspectos. A Figura 46 ilustra a página inicial de visualização dos cenários com aspectos cadastrados.

AspectNFR

Menu

Sistema
Subsistema
Caso de Uso
Requisito Não-Funcional
Atributo
Cenário
Tipo de Fluxo
Matrizes
Espalhamento de Aspectos
Cenários com Aspectos
Aspectos Identificados

Cenário com Aspecto

Nome do Sistema	Nome do Subsistema	Nome do Caso de Uso	Descrição do Cenário	
Framek	Controle Login	Conferir Login	O caso de uso verifica se os campos usuário e senha que foram informados pelo usuário correspondem as informações inseridas no banco de dados	<input type="button" value="Visualizar"/>
Framek	Controle Login	Logar no Sistema	Este caso de uso inicia quando o usuário executa a aplicação.	<input type="button" value="Visualizar"/>

Figura 46 – Página Inicial para Cenário com Aspecto

Fonte: Autoria Própria

Nesta figura são listados os cenários com aspectos cadastrados, com uma opção de *Visualizar* que permite a visualização do cenário com os aspectos identificados. A Figura 47 ilustra um exemplo de cenário com aspectos para o caso de uso *Logar no Sistema*.

Cenário com Aspecto

UC18	Logar no Sistema
Breve Descritivo: Este caso de uso inicia quando o usuário executa a aplicação.	
Fluxo Básico	
P1 - O sistema pede que o usuário informe usuário e senha	
P2 - O usuário digita seu nome usuário e sua senha no sistema e clica no botão logar <i>Join point</i>	
P3 - O sistema verifica a inserção dos dados através de Conferir Login <i>Join point</i>	
P4 - O sistema autoriza o acesso do usuário, mostrando-lhe a tela inicial do sistema com suas configurações pessoais <i>Join point</i>	
P5 - O caso de uso termina	
Fluxo Alternativo	
P1 - No passo 3 caso ocorra algum erro no preenchimento de campos obrigatórios ou na validação dos dados informados, o sistema limpará os campos e mostrará a tela de login novamente <i>Join point</i>	
Sub Fluxo	
Tipos de Aspectos	
Desempenho	

Figura 47 – Cenário com Aspecto *Logar no Sistema* gerado pela ferramenta AspectRNF

Fonte: Autoria Própria

Um cenário com aspectos é composto de seu código ilustrado por *UC18*, o nome do caso de uso, breve descrição do caso de uso, os fluxos podendo estes ser do tipo básico, alternativo ou subfluxo, os passos de cada fluxo sendo que caso o passo apresente algum aspecto cadastrado será ilustrado ao seu final um *Join Point*, e os tipos de aspecto presentes no cenário.

A ferramenta possibilita também visualizar os tipos de RNF identificados nos casos de uso cadastrados. A Figura 48 ilustra um exemplo de página inicial de seleção dos RNF.

AspectNFR

Menu

- Sistema
- Subsistema
- Caso de Uso
- Requisito Não-Funcional
- Atributo
- Cenário
- Tipo de Fluxo
- Matrizes
- Espalhamento de Aspectos
- Cenários com Aspectos
- Tipos de Aspectos**

Tipos de Aspectos

Nome do Requisito Não-Funcional	
Desempenho	Visualizar

Figura 48 – Página Inicial Aspectos Identificados por Requisito Não-Funcional

Fonte: Aatoria Própria

Nesta figura são listados RNF que possuem relacionamento com algum cenário. A opção *Visualizar* permite mostrar os passos dos casos de uso os quais estejam vinculados a algum atributo do RNF. A Figura 49 ilustra um exemplo desta visualização.

Tipos de Aspectos

Desempenho	
Conferir Login - FB P1	D1 D5 D10
Conferir Login - FB P2	D6 D10 D11 D12
Conferir Login - FB P3	D9
Logar no Sistema - FA P1	D1 D6 D9 D10 D15 D16
Logar no Sistema - FB P2	D1 D6
Logar no Sistema - FB P3	D1 D6 D9 D10 D15 D16
Logar no Sistema - FB P4	D1 D6

Figura 49 – Casos de Uso Vinculados a Atributos do RNF Desempenho

Fonte: Autoria Própria

Observa-se por meio desta figura que os casos de uso *Conferir Login* e *Logar no Sistema* contém o RNF de *Desempenho*. É possível verificar também quais são os tipos de fluxo e seus respectivos passos que apresentam os atributos relacionados a *Desempenho*.

4 TESTES E RESULTADOS

Este capítulo relata os resultados atingidos por meio deste trabalho. A Seção 4.1 apresenta exemplos de aplicação usando a ferramenta para identificação de aspectos. A Seção 4.2 descreve uma análise qualitativa entre a ferramenta proposta e as da literatura.

4.1 OUTROS EXEMPLOS DE APLICAÇÃO DA FERRAMENTA

Nesta seção são apresentados alguns exemplos mais aprofundados do uso da ferramenta para detecção de aspectos entre sistemas, subsistemas ou casos de uso. Os exemplos utilizados estão baseados no trabalho de Siqueira da Silva (2012).

Além do sistema Framemk contendo o subsistema *Controle de Login*, apresentado na Seção 3.3, foi criado o sistema *Custos de Obra* o qual apresenta como subsistema *Custos de Obra Civil* e como caso de uso *Realizar Cadastro de Obra*. O Quadro 14 apresenta a descrição textual do caso de uso *Realizar Cadastro de Obra*.

Realizar Cadastro de Obra	
Breve Descritivo:	Este Caso de Uso descreve o processo de cadastramento de obra.
Fluxo Principal	
P1- O usuário seleciona a opção de cadastrar obra.	
P2- O sistema disponibiliza um cadastro de obra para o usuário.	
P3- O usuário informa os dados.	
P4- O sistema valida os dados.	
P5- O usuário tenta salvar o cadastro.	
P6- O sistema salva o cadastro e exibe uma mensagem de sucesso.	

Quadro 14- Descrição Textual Realizar Cadastro de Obra
Fonte: Adaptado de Siqueira da Silva (2012, p. 43)

O Quadro 15 ilustra a o vínculo dos atributos do RNF *Desempenho* para o caso de uso *Realizar Cadastro de Obra*.

Cadastro de Obra								
	D ₁	D...	D ₅	D ₆	D...	D ₁₁	D...	D ₁₅
FB P ₁								
FB P ₂	X			X				
FB P ₃				X				
FB P ₄				X		X		
FB P ₅	X		X	X		X		
FB P ₆								X

Quadro 15- Matriz de Adjacência para a Descrição Textual Realizar Cadastro de Obra
Fonte: Siqueira da Silva (2012)

Dadas estas informações, foram realizados os cadastros conforme o processo descrito na seção 3.3. A Figura 50 ilustra a matriz de adjacência gerada para o caso de uso *Realizar Cadastro de Obra* pela ferramenta. Nota-se que para o fluxo básico FB2 foram detectados dois atributos do requisito não-funcional de *Desempenho*, a saber, D1 e D6. Isto é observado para os outros fluxos do caso de uso.

Matriz

Realizar Cadastro de Obra - Desempenho

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16
FB1																
FB2	X					X										
FB3						X										
FB4						X					X					
FB5	X				X	X					X					
FB6															X	

Figura 50 – Página com a Matriz de Adjacência para o Caso de Uso Realizar Cadastro de Obra

Fonte: Autoria Própria

Outra matriz gerada é a auxiliar de espalhamento. Para se gerar a matriz auxiliar pode ser dado como entrada casos de uso, subsistemas ou sistemas. A

Figura 51 ilustra a seleção de subsistemas para a geração da matriz auxiliar com espalhamento entre subsistemas para o RNF *Desempenho*.



Figura 51 – Página de Seleção de Espalhamento de Aspectos entre Subsistemas

Fonte: Autoria Própria

Nessa figura há a possibilidade de seleção dos subsistemas *Controle de Login* e *Custos de Obra Civil*. A Figura 52 ilustra a matriz auxiliar gerada para estes subsistemas.

Matriz Auxiliar

Requisito Não-Funcional: Desempenho

Subsistema: Controle Login - Custos de Obra Civil

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16
Logar no Sistema - FB2	×					×										
Logar no Sistema - FB3	×					×			×	×					×	×
Logar no Sistema - FB4	×					×										
Logar no Sistema - FA1	×					×			×	×					×	×
Conferir Login - FB1	×									×						
Conferir Login - FB2						×				×	×					
Conferir Login - FB3									×							
Realizar Cadastro de Obra - FB2	×					×										
Realizar Cadastro de Obra - FB3						×										
Realizar Cadastro de Obra - FB4						×					×					
Realizar Cadastro de Obra - FB5	×					×					×					
Realizar Cadastro de Obra - FB6															×	

Figura 52 – Matriz Auxiliar entre os Subsistemas Controle de Login e Custos de Obra Civil

Fonte: Autoria Própria

Nessa figura, acima da matriz se encontram quais os subsistemas selecionados, no caso, *Controle de Login* e *Custos de Obra Civil*, e na matriz se mostram os casos de uso com seus respectivos passos os quais estejam vinculados a um atributo do RNF *Desempenho*. Nesta matriz nota-se que o atributo D1, D6, D11 e D15 são encontrados nos dois subsistemas e D9, D10 e D16 foram identificados somente no subsistema de *Controle de Login*.

Outra matriz auxiliar que pode ser gerada é a de espalhamento entre sistemas. Não será ilustrado neste trabalho o espalhamento de aspectos entre sistemas, pois terá como resultado a mesma imagem da Figura 52, pois os subsistemas *Controle de Login* e *Custos de Obra Civil* correspondem aos sistemas Framenk e Custos de Obra, gerando o mesmo espalhamento.

4.2 ANÁLISE DAS FERRAMENTAS SIMILARES

Esta seção apresenta uma análise qualitativa da ferramenta desenvolvida com as presentes na literatura.

A ferramenta Theme/Doc utiliza a abordagem baseada em temas provendo vistas e suporte de funcionalidades para a identificação e representação de aspectos na fase de análise. O Quadro 16 ilustra as principais diferenças entre a ferramenta proposta e a ferramenta Theme/Doc.

Crítérios de Comparação	Ferramenta Proposta	Theme/Doc
Identificação de Aspectos	A partir de RNFs	Baseado em temas
Análise	Descrição textual para formar os cenários e análise dos passos da descrição para identificação dos atributos relacionados	Documento de requisitos, separando palavras do tipo chave para formar o conjunto de temas.
Artefatos gerados pela Ferramenta	Matriz de espalhamento, cenário com passos em que ocorrem pontos de junção e RNFs identificados dentro do cenário e visualização do RNF espalhado entre os cenários e quais atributos relacionados a quais passos.	Vistas de ação e vista de ação <i>clipped</i> de forma automatizada facilitando a identificação de temas <i>crosscutting</i> e base
Limitações	Identificação de atributos de RNF nos passos do caso de uso devem ser identificados pelo analista.	O conjunto de temas necessita ser refinado de forma iterativa e intuitiva
Identificação de Aspectos entre sistemas, subsistemas e caso de uso	Sim	Somente entre requisitos

Quadro 16– Comparativo entre a Ferramenta Proposta e a Ferramenta Theme/Doc de Baniassad e Clarke (2004)
Fonte: Aatoria Própria

A principal diferença entre as ferramentas está na forma em como são identificados os aspectos. A ferramenta Theme/Doc propõe a identificação através da abordagem por temas, sendo geradas as vistas de ação e *clipped* para o auxílio na identificação de temas *crosscutting* e base. Já a ferramenta proposta se utiliza de cenários e o relacionamento entre passos e atributos de RNF para a identificação de aspectos.

Outra ferramenta é a Minerando Aspectos em Requisitos a qual apresenta uma forma de identificação de candidatos a aspectos através do

processamento de descrições textuais. O Quadro 17 ilustra as principais diferenças entre esta ferramenta e a proposta.

Critérios de comparação	Ferramenta Proposta	Minerando Aspectos em Requisitos
Forma de Análise	Manual	Através de um processador de linguagem natural
Artefatos gerados pela Ferramenta	Matriz de espalhamento, cenário com aspecto e RNFs identificados	Pontos de vista e requisitos transversais
Identificação de Aspectos	Com base em RNF	Nem sempre é capaz de identificar aspectos a partir de RNFs
Identificação de Aspectos entre sistemas, subsistemas e caso de uso	Sim	A partir de relacionamentos transversais, mas não entre sistemas e subsistemas

Quadro 17– Comparativo entre a Ferramenta Proposta e a Minerando Aspectos em Requisitos por Sampaio et al. (2005a)
Fonte: Aatoria Própria

Na ferramenta Minerando Aspectos em Requisitos é utilizado um processador de linguagem natural para a análise das descrições textuais, e após esta análise são identificadas as frequências com que verbos de ação e substantivos ocorrem. Os verbos identificados com maior frequência podem ser traduzidos para requisitos transversais e os substantivos com maior frequência podem tornar-se pontos de vista.

A principal diferença entre as duas ferramentas está na abordagem de identificação de aspectos, em que a ferramenta Minerando Aspectos em Requisitos leva em consideração todo o documento de requisitos para auxiliar na identificação dos requisitos transversais, enquanto que a ferramenta proposta apesar de analisar descrições textuais, seu foco está na identificação de RNF os quais estão presentes nas descrições textuais. Lembra-se que os aspectos apresentam diversas similaridades com RNFs, portanto uma das formas ideais de se identificar aspectos é considerando RNFs (Siqueira da Silva, 2012).

Outro ponto relevante é que a ferramenta proposta além de identificar aspectos entre casos de uso, também é capaz de identificá-los entre sistemas e subsistemas. Esta característica não é encontrada nas ferramentas Theme/Doc e Minerando Aspectos em Requisitos.

5 CONCLUSÃO

Este trabalho apresentou uma ferramenta com base no método proposto por Siqueira da Silva (2012) capaz de identificar aspectos a partir de atributos de RNFs durante a fase de análise.

O processo implementado na ferramenta AspectRNF é capaz de identificar os aspectos espalhados entre sistemas, subsistemas e casos de uso. Esta identificação foi incorporada ao método de Siqueira da Silva (2012), pois o mesmo não a contemplava.

A modelagem lógica da AspectRNF foi feita utilizando o diagrama de classes UML e a modelagem de banco de dados usou o Modelo Entidade Relacionamento.

O processo desenvolvido pode iniciar por dois cadastros: sistema/subsistema ou de RNF. Caso o cadastro seja sistema/subsistema, é realizado a inserção de requisitos funcionais (casos de uso). Do contrário, realiza-se o cadastro dos atributos de RNF.

Depois desses cadastros são informados os cenários e sua associação com os atributos de RNF. Em seguida, podem ser geradas as matrizes adjacência de cada caso de uso. A matriz auxiliar para verificar o espalhamento de aspectos entre sistemas, subsistemas ou casos de uso também está disponível. Podem-se gerar o cenário com aspectos e a visualização dos passos de caso de uso que estão vinculados a um RNF.

O funcionamento da ferramenta foi testado por meio de exemplos contemplados em Siqueira da Silva (2012) para o sistema FrameMK e Controle de Obras. Observou-se que a ferramenta produziu os mesmos resultados apresentados nesse trabalho, validando o processo implementado.

Também foram apresentadas outras duas ferramentas, a ferramenta *Theme/Doc* e a *Minerando Aspectos em Requisitos* presentes na literatura as quais são capazes de identificar aspectos na fase de análise, e se realizou uma análise qualitativa entre estas e a ferramenta proposta, ilustrando suas principais diferenças. A ferramenta *Theme/Doc* possui por característica a análise de temas para a identificação de aspectos, a *Minerando Aspectos em Requisitos* realiza a identificação de aspectos analisando requisitos transversais em descrições textuais. Por sua vez, a AspectRNF realiza a

identificação de aspectos a partir da análise de atributos de RNF vinculados a passos de casos de uso e identifica aspectos espalhados entre sistemas, subsistemas e casos de uso.

5.1 TRABALHOS FUTUROS

Um trabalho futuro que pode ser desenvolvido é a refatoração de código da ferramenta para melhorar sua manutenibilidade.

Um segundo trabalho é refinar os atributos de RNFs encontrados no trabalho de Mairizia et al. (2010), eliminando atributos que não poderão ser transformados em aspectos em fases posteriores do desenvolvimento de software.

Outro trabalho é a possibilidade de importação e exportação de casos de uso na ferramenta.

A criação de um processo para a fase de projeto orientado a aspectos é outro trabalho de pesquisa que pode ser desenvolvido. Esse processo receberia como entrada os atributos de RNFs identificados, gerando como saída diagramas que representa esses atributos dentro do sistema a ser implementado. Por fim, realizar a inserção deste processo na ferramenta AspectRNF.

REFERÊNCIAS

BAKKER, Jethro; TEKINERDOGAN, Bedir; AKSIT Mehmet. Characterization of early aspects approaches; **Proceedings...** Early Aspects Workshop at AOSD'05, 2005.

BANIASSAD, Elisa; CLARKE, Siobhán. Theme: An approach for aspect-oriented analysis and design. **Proceedings...** International Conference on Software Engineering, 2004.

CLARKE, Siobhán; BANIASSAD, Elisa. **Aspect-Oriented Analysis and Design – The Theme Approach**. Addison-Wesley, Boston, 2005.

HORNUNG, Rafael. **Um método para identificação antecipada de candidatos a aspectos no desenvolvimento de frameworks de domínio**. 2010. 83f. Dissertação (Mestrado em Informática) - Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná. Curitiba, 2010.

JACOBSON, Ivar; NG, Pan-Wei. **Aspect-Oriented Software Development with Use Cases**. Addison-Wesley, Boston, 2004.

KOSSOSKI, Clayton. **Modelagem de Sistemas Baseada em Orientação a Aspectos Usando Casos de Uso e Responsabilidades**. Trabalho de Conclusão de Curso - Curso Superior em Análise e Desenvolvimento de Sistemas - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2010.

LARMAN, C. **Utilizando UML e Padrões: Uma Introdução à análise e ao projeto orientados a objetos**. Bookman, 2007. 696p.

MAIRIZIA, Dewi; ZOWGHI, Didar; NURMULIANI, Nurie. An Investigation Into the Notion of Non-Functional Requirements. **Proceedings...** Acm Symposium on Applied Computing, Sierre, 2010.

UML. OMG Unified Modeling Language (UML), Superstructure. OMG Document Number: formal/2011-08-06. Disponível em: <
<http://www.omg.org/spec/UML/2.0/>>. Acesso em: 10 de Dezembro de 2013.

RASHID, Awais; SAWYER, Peter; MOREIRA, Ana; ARAUJO, João. Early Aspects: a Model for Aspect-Oriented Requirements Engineering. **Proceedings...** International Conference on Requirements Engineering (RE), Essen, 2002.

ROSENHAINER, Lars. The DISCERN Method: Dealing Separately with Crosscutting Concerns. **Proceedings...** OOPSLA Early Aspects 2005. San Diego, (2004).

SAMPAIO, Américo; LOUGHRAN, Neil; RASHID, Awais; RAYSON, Paul. Mining Aspects in Requirements. **Proceedings...** Early Aspects 2005 (AOSD 2005), Chicago, 2005a.

SAMPAIO, Américo; RASHID, Awais; RAYSON, Paul. Early-AIM: An Approach for Identifying Aspects in Requirements. **Proceedings...** 13th IEEE International Conference on Requirements Engineering Conference, Paris, 2005b.

SIQUEIRA DA SILVA, Leandro. **Um método para identificação de aspectos em nível de análise baseado em atributos de requisitos não-funcionais.** 2012. 92f. Trabalho de Conclusão de Curso - Curso Superior em Análise e Desenvolvimento de Sistemas - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2012.

Spring Framework. The Spring Framework – Reference Documentation. Disponível em: < <http://docs.spring.io/spring/docs/2.0.x/reference/index.html> >. Acesso em: 10 de Janeiro de 2014.

XAVIER, Laís. **Integração de Requisitos Não-Funcionais a Processos de Negócios:** Integrando BPMN e RNF. 2009. 100 f. Dissertação (Mestrado em Ciência da Computação) - Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Pernambuco, 2009.