

## Integrando Requisitos Não Funcionais à Modelagem Orientada a Objetos

Jaime de Melo Sabát Neto\*  
Julio Cesar Sampaio do Prado Leite\*\*

Departamento de Informática, PUC-Rio  
R. Marquês de São Vicente, 225  
22453-900 - Rio de Janeiro, Brasil  
e-mail: { jaime, julio }@inf.puc-rio.br

### Resumo

Recentemente, tem-se ressaltado que a maioria dos métodos da engenharia de requisitos não consideram requisitos não funcionais (RNFs) [11][12]. Conseqüentemente, ocorrem vários problemas durante o desenvolvimento de sistemas de software como, por exemplo, custos elevados e não cumprimento de prazos. Objetivando atenuar a negligência de RNFs e suas conseqüências, propõe-se, neste trabalho, uma estratégia que acrescenta os RNFs à modelagem orientada a objetos. A estratégia proposta parte de um Léxico Ampliado da Linguagem do Universo de Informações (LAL do Udi) e produz um diagrama de classes com indicações sobre quais classes, atributos, operações e relacionamentos satisfazem RNFs. Além disso, a estratégia proposta fornece suporte à elicitação de RNFs através de um Léxico de RNFs (LAL RNF), o qual registra informações sobre RNFs e possibilidades de satisfação de RNFs.

**Palavras-chave:** Requisitos Não Funcionais, Léxico Ampliado da Linguagem, Cenários, Orientação a objetos

### 1. Introdução

O mundo atual é, cada vez mais, dependente de sistemas de software. Várias funções vitais da sociedade dependem destes sistemas como, por exemplo, a telefonia, o transporte e o fornecimento de energia. Caso sejam de baixa qualidade, estes sistemas de software podem acarretar perdas de vidas humanas, danos ao ambiente e prejuízos econômicos [1].

Os atributos de qualidade de sistemas de software foram primeiramente apresentados por McCall [2] e Boehm [3]. Atualmente, o padrão internacional ISO 9126 indica que um sistema de software de qualidade deve possuir as seguintes características [4]: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade. Por outro lado, Yeh foi o primeiro a explicitar o conceito de requisitos não funcionais (RNFs) [5], os quais são atributos de qualidade ou restrições aos sistemas de software, despertando a comunidade científica para a necessidade de considerar os atributos de qualidade de sistemas de software (ou requisitos não funcionais) na fase inicial do desenvolvimento destes sistemas. Em seguida, Roman incluiu os RNFs em sua taxonomia de questões da engenharia de requisitos [7], corroborando a visão de Yeh sobre a consideração de atributos de qualidade de sistemas de software na fase de definição destes sistemas. Brackett também apresenta o conceito de RNFs no *curriculum module* intitulado

\* Apoio da Capes

\*\* Apoio do CNPq

*Software Requirements*, o qual foi desenvolvido para o *Software Engineering Institute* e tem a pretensão de servir de base para qualquer curso sobre *requisitos de software* [8].

Apesar dos RNFs terem sido apresentados pela primeira vez há algum tempo e de já fazerem parte do roteiro de ensino de requisitos, sabe-se muito pouco a respeito destes quando comparados com requisitos funcionais e outros fatores menos críticos do desenvolvimento de sistemas de software [9]. A maioria dos métodos da engenharia de software em geral, e da engenharia de requisitos em particular, não considera RNFs [5][7][10][11][12]. Várias são as causas para esta desconsideração de RNFs. Entre estas causas, destacam-se [7][12]: a existência de uma variedade de RNFs; alguns RNFs (ex. performance) estão relacionados a soluções de desenho não conhecidas na fase de definição de requisitos; outros RNFs (ex. usabilidade) são altamente subjetivos e podem ser definidos apenas através de avaliações empíricas complexas; os RNFs tendem a se relacionar com requisitos funcionais; os RNFs tendem a entrar em conflito com outros RNFs; a distinção imprecisa entre RNFs e requisitos funcionais.

As conseqüências da desconsideração de um RNF são freqüentemente mais severas do que as decorrentes da omissão de um requisito funcional [11]. Como conseqüências da desconsideração de RNFs, tem-se: aumento dos custos de desenvolvimento de sistemas de software; não cumprimento de prazos; desativação de sistemas de software (vide o caso do Sistema de Serviços de Ambulâncias de Londres [13]); comprometimento da completeza dos requisitos definidos; satisfação parcial de clientes/usuários de sistemas de software.

Cientes do fato de que a maioria dos métodos da engenharia de software e da engenharia de requisitos não consideram RNFs, como também das causas e conseqüências deste fato, propomos, neste trabalho, uma estratégia (OORNF) que permite a integração de RNFs à modelagem orientada a objetos. Em outras palavras, propomos uma estratégia orientada a objetos que fornece suporte à elicitação e modelagem de RNFs e estratégias de satisfação. Na Seção 2, apresentam-se os conceitos necessários ao entendimento da estratégia proposta. Na Seção 3, descreve-se a estratégia proposta, juntamente com exemplos extraídos da definição de um sistema de financiamento com alçada (SFA). Na Seção 4, mostram-se alguns trabalhos relacionados. Na Seção 5, apresentam-se algumas contribuições da proposta em questão e os possíveis trabalhos futuros.

## 2. Conceitos Básicos

### 2.1 – Requisitos Não Funcionais

Requisitos não funcionais (RNFs) são atributos de qualidade ou restrições de sistemas de software ou de processos de desenvolvimento de sistemas de software. Os RNFs podem ser classificados em primários ou específicos [22]. RNFs primários (ex. precisão) apresentam um alto nível de abstração e podem ser decompostos em RNFs específicos. RNFs específicos (ex. precisão de valor) apresentam um nível maior de detalhe e mostram aspectos de RNFs primários.

Estratégias de satisfação (ex. validação) são maneiras de satisfazer RNFs, identificadas na literatura ou em experiências de desenvolvimento de sistemas de software. No presente trabalho, satisfação de RNFs é entendida como satisfação dentro de limites aceitáveis [9].

Em geral, estratégias de satisfação são requisitos funcionais que podem satisfazer RNFs e também podem influir negativamente na satisfação de outros RNFs. Estas influências negativas são utilizadas para inferir conflitos entre RNFs. Por exemplo, a funcionalidade de recortar e colar, presente na maioria dos softwares, pode ser encarada como uma estratégia de satisfação que satisfaz o RNF usabilidade. Por outro lado, esta estratégia pode influir negativamente na

satisfação do RNF precisão, pois geralmente se esquece de mudar as informações copiadas e coladas. A partir desta influência negativa, infere-se que o RNF usabilidade pode entrar em conflito com o RNF precisão.

### 2.2 – Léxico Ampliado da Linguagem do Universo de Informações (LAL do UdI)

Universo de informações (UdI) é o contexto geral onde o software deverá ser desenvolvido e operado [1]. O LAL do UdI descreve a linguagem utilizada pelos atores e fontes de informação do UdI [15]. O LAL do UdI é formado por entradas, as quais descrevem símbolos da linguagem do UdI, através de noções e impactos. Estas entradas podem ser classificadas como sujeito, verbo, objeto ou estado. Para cada classe de entrada, existem heurísticas correspondentes que indicam quais informações devem ser registradas em suas noções e impactos. Ao descrever entradas do LAL do UdI, deve-se seguir os princípios de vocabulário mínimo e de circularidade. O princípio de vocabulário mínimo prescreve a minimização da utilização de símbolos externos ao LAL do UdI na descrição de entradas do LAL do UdI. O princípio de circularidade prescreve a maximização da utilização de símbolos do LAL do UdI na descrição de entradas do LAL do UdI. Como decorrência do princípio de circularidade, tem-se que o LAL do UdI é um hipertexto. As entradas do LAL do UdI são os nós do hipertexto. Os símbolos, que aparecem nas descrições de entradas, são os elos do hipertexto. A Figura 1 mostra as entradas *conferir ficha cadastral* e *ficha cadastral*, classificadas respectivamente como verbo e objeto, pertencentes ao LAL do UdI de um SFA.

<p><b>Conferir ficha cadastral/Confere ficha cadastral/Conferir contrato</b></p> <p><b>Noções</b></p> <ul style="list-style-type: none"> <li>Tarefa realizada pelo <i>loja</i> que consiste em verificar se as informações da <i>ficha cadastral</i> estão corretas.</li> <li>Acomece depois que <i>cliente</i> entrega <i>ficha cadastral</i> e apresenta <i>carteira de identidade</i> e <i>CNPJ</i> para <i>loja</i>.</li> </ul> <p><b>Impactos</b></p> <ul style="list-style-type: none"> <li><i>Loja</i> preenche os campos não informados pelo <i>cliente</i> com a sigla "NE" (não existente).</li> <li><i>Loja</i> confere número de <i>identidade</i> presente na <i>ficha cadastral</i> com a <i>carteira de identidade</i> do <i>cliente</i>.</li> <li><i>Loja</i> confere número de <i>CNPJ</i> presente na <i>ficha cadastral</i> com o <i>CNPJ</i> do <i>cliente</i>.</li> <li><i>Loja</i> confere telefone de contato e endereço presentes na <i>ficha cadastral</i>, ligando para o telefone de contato.</li> </ul>	<p><b>Ficha cadastral/Proposta de crédito/Contrato/Contratos</b></p> <p><b>Noções</b></p> <ul style="list-style-type: none"> <li>Ficha preenchida pelo <i>cliente</i> que deseja ter uma <i>aprovação de crédito</i> para fazer compra em uma <i>loja</i>.</li> <li>Possui <i>número de proposta</i>, <i>número de contrato</i>, <i>número de identidade</i>, <i>número de CNPJ</i>, <i>nome do cliente</i>, <i>nome do banco</i>, <i>endereço</i>, <i>telefone de contato</i>, <i>valor da prestação</i> e <i>número de prestações</i>.</li> <li>Esta associada a <i>cheques</i>.</li> <li>Pode ser uma <i>pendência</i>.</li> </ul> <p><b>Impactos</b></p> <ul style="list-style-type: none"> <li><i>Cliente</i> preenche <i>ficha cadastral</i>.</li> <li><i>Cliente</i> entrega <i>ficha cadastral</i> para a <i>loja</i>.</li> <li><i>Cliente</i> assina <i>ficha cadastral</i>.</li> <li><i>Loja</i> cria <i>contratos</i> para <i>financiamento</i>, no <i>enviar movimento</i>.</li> <li><i>Interface</i> marca <i>contrato</i> como <i>pendência</i>.</li> <li><i>Financiamento</i> marca <i>contrato</i> como <i>não pendência</i>.</li> </ul>
---	--

Figura 1. Entradas *Conferir ficha cadastral* e *ficha cadastral* de um LAL do UdI

### 2.3. Léxico Ampliado da Linguagem de RNFs / Base de Conhecimento de RNFs

O Léxico Ampliado da Linguagem de RNFs (LAL RNF) registra a linguagem, associada a RNFs, que deveria ser utilizada por engenheiros de software/engenheiros de requisitos [22]. Os símbolos desta linguagem são RNFs primários, RNFs específicos e estratégias de satisfação. O

<p><b>nome do RNF primário</b></p> <p><b>Noções</b></p> <ul style="list-style-type: none"> <li>descrição do RNF primário</li> <li>Possui <math>\alpha(s)</math> seguinte(s) RNF(s) específico(s): RNF específico 1, RNF específico 2, ... e RNF específico n.</li> <li>Possui <math>\alpha(s)</math> seguinte(s) estratégia(s) de satisfação: estratégia de satisfação 1, estratégia de satisfação 2, ... e estratégia de satisfação n.</li> </ul> <p><b>Impactos</b></p> <ul style="list-style-type: none"> <li>Pode entrar em conflito com <math>\alpha(s)</math> seguinte(s) RNF(s) primário(s): RNF primário 1, RNF primário 2, ... e RNF primário n.</li> </ul>	<p><b>nome do RNF específico</b></p> <p><b>Noções</b></p> <ul style="list-style-type: none"> <li>descrição do RNF específico</li> <li>Esta relacionado ao RNF primário RNF primário</li> <li>Possui <math>\alpha(s)</math> seguinte(s) estratégia(s) de satisfação: estratégia de satisfação 1, estratégia de satisfação 2, ... e estratégia de satisfação n.</li> </ul> <p><b>Impactos</b></p> <ul style="list-style-type: none"> <li>Pode entrar em conflito com <math>\alpha(s)</math> seguinte(s) RNF(s) específico(s): RNF específico 1, RNF específico 2, ... e RNF específico n.</li> </ul>	<p><b>nome da estratégia de satisfação</b></p> <p><b>Noções</b></p> <ul style="list-style-type: none"> <li>descrição da estratégia de satisfação</li> </ul> <p><b>Impactos</b></p> <ul style="list-style-type: none"> <li>Influi positivamente na satisfação do RNF específico RNF específico 1, comentário sobre a influência.</li> <li>Influi positivamente na satisfação do RNF específico RNF específico n, comentário sobre a influência.</li> <li>Influi negativamente na satisfação do RNF específico RNF específico 1, comentário sobre a influência.</li> <li>Influi negativamente na satisfação do RNF específico RNF específico n, comentário sobre a influência.</li> </ul>
--	--	---

Figura 2. Templates para entradas do LAL RNF

LAL RNF, assim como o LAL do Udi, é composto por entradas, deve seguir os princípios de circularidade e vocabulário mínimo e pode ser apresentado como um hipertexto. No entanto, as entradas do LAL RNF são classificadas como RNF primário, RNF específico ou estratégia de satisfação. Para cada tipo de entrada do LAL RNF, existe um template (Figura 2) que indica o que deve ser registrado nas noções e impactos da entrada. Como as entradas do LAL RNF descrevem RNFs e estratégias de satisfação, o LAL RNF pode ser encarado como uma *base de conhecimento de RNFs*. Deste modo, o LAL RNF fornece apoio à elicitação de RNFs e estratégias de satisfação.

### 2.4. Cenários

Neste trabalho, utiliza-se a proposta de Leite para cenários [17]. Os pontos centrais desta proposta são: cenários descrevem situações que acontecem no macrosistema; cenários evoluem durante o desenvolvimento de software; cenários estão naturalmente ligados ao LAL do Udi e cenários são descritos em linguagem natural. Cada cenário possui título, objetivo, contexto, atores, recursos e episódios. Para recursos e episódios, pode-se registrar restrições. As restrições de episódios registram as estratégias de satisfação especializadas pelos episódios e os RNFs específicos satisfeitos pelos episódios. As restrições de recursos registram RNFs específicos que restringem ou indicam atributos de qualidade para os recursos. A Figura 3 apresenta o cenário *Loja confere ficha cadastral*, o qual ocorre em um SFA.

<b>Título:</b> Loja confere ficha cadastral
<b>Objetivo:</b> Verificar se as informações da <u>ficha cadastral</u> estão corretas
<b>Contexto:</b> Cliente entrega <u>ficha cadastral</u> e apresenta <u>carteira de identidade</u> e <u>CPE</u> para loja
<b>Atores:</b> loja
<b>Recursos:</b> <u>carteira de identidade</u> , <u>CPE</u> , <u>ficha cadastral</u> Restrição: deve ter <u>precisão de propriedade</u> e <u>precisão de valor</u> .
<b>Episódios</b>
Loja preenche os campos da <u>ficha cadastral</u> não informados pelo <u>cliente</u> com a sigla 'NE' (não existente)
Restrição: <u>ficha cadastral</u> deve ter <u>precisão de propriedade</u> , sendo a estratégia de satisfação <u>validação</u>
Loja verifica o número da <u>identidade</u> presente na <u>ficha cadastral</u> com a <u>carteira de identidade</u> do <u>cliente</u>
Restrição: <u>ficha cadastral</u> deve ter <u>precisão de valor</u> , sendo a estratégia de satisfação <u>validação</u>
Loja verifica o número de <u>CPE</u> presente na <u>ficha cadastral</u> com o <u>CPE</u> do <u>cliente</u>
Restrição: <u>ficha cadastral</u> deve ter <u>precisão de valor</u> , sendo a estratégia de satisfação <u>validação</u>
Loja verifica o endereço e o telefone de contato presentes na <u>ficha cadastral</u> , ligando para o <u>telefone de contato</u>
Restrição: <u>ficha cadastral</u> deve ter <u>precisão de valor</u> , sendo a estratégia de satisfação <u>validação</u> .

Figura 3. Cenário Loja confere ficha cadastral

### 3. Estratégia OORNF

A estratégia OORNF parte de um Léxico Ampliado da Linguagem do Universo de Informações

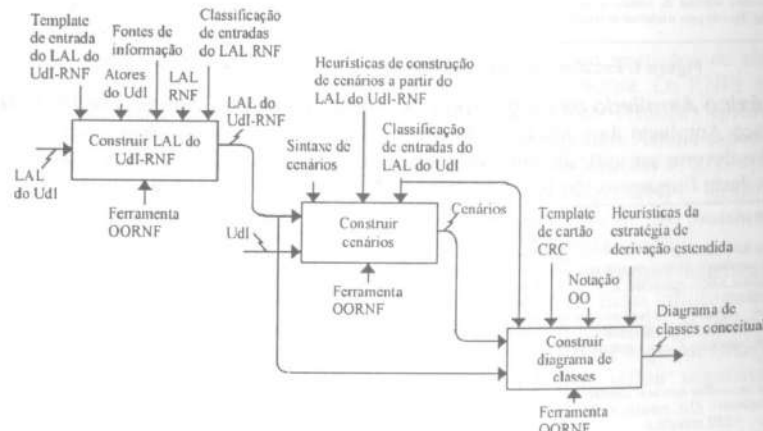


Figura 4. Diagrama SADT para a estratégia OORNF

(LAL do Udi) e produz um diagrama de classes conceitual com indicações sobre quais classes, atributos, operações e relacionamentos especializam estratégias de satisfação e satisfazem RNFs. Por conceitual, indica-se que o diagrama de classes apresenta apenas classes semânticas, identificadas no espaço do problema/Udi [14]. A estratégia OORNF engloba as seguintes atividades: construir LAL do Udi-RNF, construir cenários e construir diagrama de classes. A Figura 4 apresenta um diagrama SADT para a estratégia OORNF.

### 3.1. Construir LAL do Udi-RNF

Nesta atividade, constrói-se o Léxico Ampliado da Linguagem do Udi-RNF (LAL do Udi-RNF) a partir do LAL do Udi. O LAL do Udi-RNF é um LAL do Udi que possui registro sobre quais entradas, noções e impactos especializam estratégias de satisfação e satisfazem RNFs primários e RNFs específicos. Estes RNFs primários, RNFs específicos e estratégias de satisfação são descritos por entradas do LAL RNF, as quais passam a ser entradas do LAL do Udi-RNF. As entradas, noções e impactos, que especializam estratégias de satisfação e satisfazem RNFs, podem estar presentes no LAL do Udi ou serem elicidadas durante a construção do LAL do Udi-RNF.

A atividade construir LAL do Udi-RNF é composta pelos seguintes passos: verificar entradas que especializam estratégias de satisfação e satisfazem RNFs, verificar impactos que especializam estratégias de satisfação e satisfazem RNFs e identificar novas entradas, noções e impactos. Descreve-se abaixo o processo de construção do LAL do Udi-RNF. Mostram-se alguns exemplos, utilizando, como entrada para o processo, o LAL do Udi de um SFA. As palavras sublinhadas são símbolos do LAL do Udi-RNF.

1. Recuperar todas entradas do LAL RNF classificadas como estratégia de satisfação, RNF primário e RNF específico.

**Exemplo:** Ao examinar o LAL RNF, recuperaram-se as entradas da classe estratégia de satisfação como, por exemplo: *alarme*, *auditoria de precisão*, *autenticação*, *aviso de imprecisão*, *confirmação*, *faixa de valores*, *identificação*, *informações de apoio*, *validação* (Figura 5) e *verificação de consistência*. Recuperaram-se também as entradas da classe RNF

<p><b>precisão</b></p> <p><b>Noções</b></p> <ul style="list-style-type: none"> <li>• No contexto de sistemas de informação, refere-se à fidelidade dos itens de informação em relação ao que representam no Udi.</li> <li>• Possui o(s) seguinte(s) RNF(s) específico(s): <u>consistência externa</u>, <u>consistência interna</u>, <u>precisão de oportunidade</u>, <u>precisão de propriedade</u>, <u>precisão de valor</u> e <u>precisão um para um</u>.</li> <li>• Possui n(s) seguinte(s) estratégia(s) de satisfação: <u>auditoria de precisão</u>, <u>autorização</u>, <u>aviso de imprecisão</u>, <u>aviso preventivo de imprecisão</u>, <u>validação</u>, <u>capacidade de execução</u>, <u>confirmação</u>, <u>faixa de valores</u>, <u>impressão de qualidade</u>, <u>informações de apoio</u>, <u>validação</u>, <u>validação automática</u>, <u>validação na fonte</u> e <u>verificação de consistência</u>.</li> </ul> <p><b>Impactos</b></p> <ul style="list-style-type: none"> <li>• Pode entrar em conflito com o(s) seguinte(s) RNF(s) específico(s): <u>confidencialidade</u>, <u>custo operacional</u>, <u>performance de acesso</u>, <u>precisão de oportunidade</u> e <u>rapidez de utilização</u>.</li> </ul>	<p><b>precisão de valor</b></p> <p><b>Noções</b></p> <ul style="list-style-type: none"> <li>• Item de informação devem ter valores corretos.</li> <li>• Está relacionado ao RNF primário <u>precisão</u>.</li> <li>• Possui n(s) seguinte(s) estratégia(s) de satisfação: <u>auditoria de precisão</u>, <u>autorização</u>, <u>aviso de imprecisão</u>, <u>aviso preventivo de imprecisão</u>, <u>capacidade de execução</u>, <u>confirmação</u>, <u>faixa de valores</u>, <u>informações de apoio</u>, <u>validação</u>, <u>validação automática</u>, <u>validação na fonte</u> e <u>verificação de consistência</u>.</li> </ul> <p><b>Impactos</b></p> <ul style="list-style-type: none"> <li>• Pode entrar em conflito com o(s) seguinte(s) RNF(s) específico(s): <u>confidencialidade</u>, <u>custo operacional</u>, <u>performance de acesso</u>, <u>precisão de oportunidade</u> e <u>rapidez de utilização</u>.</li> </ul>	<p><b>validação</b></p> <p><b>Noções</b></p> <ul style="list-style-type: none"> <li>• Um validador, diferente do emissor, inspeciona itens de informação, utilizando <u>informações de apoio</u> e orientações procedimentais, para garantir que os itens de informação obedecem a padrões pre-determinados.</li> </ul> <p><b>Impactos</b></p> <ul style="list-style-type: none"> <li>• Influê positivamente na satisfação do RNF específico <u>precisão de valor</u>, pois a inspeção das informações por uma pessoa diferente do emissor aumenta a possibilidade de descobrir valores incorretos destas informações.</li> <li>• Influê positivamente na satisfação do RNF específico <u>precisão de propriedade</u>, pois a inspeção das informações por uma pessoa diferente do emissor aumenta a possibilidade de descobrir propriedades desejadas destas informações, que não estão presentes.</li> <li>• Influê negativamente na satisfação do RNF específico <u>precisão de oportunidade</u>, caso seja necessário um tempo excessivo para a <u>validação</u>.</li> <li>• Influê negativamente na satisfação do RNF específico <u>confidencialidade</u>, caso o validador não possa ter acesso as informações validadas.</li> </ul>
--	---	--

Figura 5. Entradas *precisão*, *precisão de valor* e *validação* do LAL RNF

primário como, por exemplo: *custo*, *performance*, *precisão* (Figura 5), *segurança* e *usabilidade*. Finalmente, recuperaram-se as entradas do LAL RNF da classe RNF específico como, por exemplo: *confidencialidade*, *custo de desenvolvimento*, *custo operacional*, *disponibilidade*, *facilidade de aprendizado*, *performance de espaço*, *precisão de oportunidade*, *precisão de propriedade*, *precisão de valor* (Figura 5) e *rapidez de utilização*.

Para cada entrada do LAL do Udi, faça:

2. Verificar se a entrada do LAL do Udi especializa estratégia de satisfação e satisfaz RNFs, utilizando as entradas do LAL RNF. Caso a entrada do LAL do Udi especialize uma estratégia de satisfação e satisfaça RNFs, incluir noções para a entrada em questão que registram estas informações. Nesta situação, também são incluídos impactos para a entrada do LAL do Udi que indicam conflitos com RNFs.

**Exemplo:** Ao percorrer as entradas do LAL RNF classificadas como estratégia de satisfação e verificar a entrada *Conferir ficha cadastral* do LAL do Udi (Figura 1), percebeu-se que a entrada *Conferir ficha cadastral* especializava a entrada do LAL RNF *validação* (Figura 5). Incluiu-se a seguinte noção para a entrada *conferir ficha cadastral* do LAL do Udi-RNF (Figura 7): *É uma especialização da estratégia de satisfação validação*. A entrada do LAL RNF *validação* passou a ser uma entrada do LAL do Udi-RNF. Ao verificar os impactos da entrada do LAL RNF *validação* (Figura 5), identificaram-se os seguintes RNFs específicos que são influenciados positivamente pela estratégia: *precisão de valor* e *precisão de propriedade*. Escolheram-se os dois RNFs específicos como sendo aqueles que a entrada *Conferir ficha cadastral* pretende satisfazer. Ao examinar as noções das entradas do LAL RNF referentes a *precisão de valor* (Figura 5) e *precisão de propriedade*, identificou-se *precisão* como sendo o RNF primário destes RNFs específicos. Incluíram-se as seguintes noções para a entrada *Conferir ficha cadastral* (Figura 7): *Objetiva satisfazer o(s) seguinte(s) RNF(s) primário(s): precisão* e *Objetiva satisfazer o(s) seguinte(s) RNF(s) específico(s): precisão de valor e precisão de propriedade*. As entradas do LAL RNF *precisão*, *precisão de valor* e *precisão de propriedade* passaram a ser entradas do LAL do Udi-RNF. Ao verificar os impactos da entrada *validação* do LAL RNF, identificaram-se os seguintes RNFs específicos que são influenciados negativamente pela estratégia: *confidencialidade* e *precisão de oportunidade*. Ao verificar as noções das entradas do LAL RNF associadas a estes RNFs específicos, identificaram-se *segurança* e *precisão* como seus respectivos RNFs primários. Incluíram-se os seguintes impactos para a entrada *Conferir ficha cadastral* (Figura 7): *Pode entrar em conflito com o(s) seguinte(s) RNF(s) primário(s): segurança e precisão* e *Pode entrar em conflito com o(s) seguinte(s) RNF(s) específico(s): confidencialidade e precisão de oportunidade*. As entradas do LAL RNF *segurança*, *precisão*, *confidencialidade* e *precisão de oportunidade* passaram a ser entradas do LAL do Udi-RNF.

Para cada noção da entrada do LAL do Udi, faça:

- 2.1. Verificar se a noção especializa estratégia de satisfação e satisfaz RNFs, utilizando as entradas do LAL RNF. Caso a noção especialize uma estratégia de satisfação e satisfaça RNFs, registrar esta informação, entre parêntesis, ao final da noção.

Loja/Lojas/Lojista/Lojistas
<b>Noções</b>
<ul style="list-style-type: none"> <li>Loja que possui um acordo com a <i>financeira</i> para a concessão de financiamento a seus <i>clientes</i>.</li> <li>Possui uma <i>alçada</i>, a qual é definida pela <i>financeira</i>.</li> <li>Possui um código de acesso para utilizar a <i>interface</i> (<i>identificação satisfazendo confidencialidade</i>).</li> <li>Possui uma senha para utilizar a <i>interface</i> (<i>autenticação satisfazendo confidencialidade</i>).</li> </ul>
<b>Impactos</b>
<ul style="list-style-type: none"> <li>Loja <i>confere</i> <i>ficha cadastral</i> (<i>validação satisfazendo precisão de valor e precisão de propriedade</i>).</li> <li>Loja <i>aprova</i> <i>crédito</i>.</li> <li>Loja <i>não aprova</i> <i>crédito</i>.</li> <li>Loja <i>cria</i> <i>movimento para financeira</i>.</li> <li>Loja <i>recebe</i> código de acesso e senha da <i>financeira</i> para utilizar a <i>interface</i>.</li> </ul>

Figura 6. Entrada *Loja* de um LAL do Udi-RNF

**Exemplo:** Ao percorrer as entradas do LAL RNF classificadas como estratégia de satisfação, percebeu-se que a noção *Possui um código de acesso para utilizar a interface* da entrada do LAL do Udi *Loja* especializava a entrada do LAL RNF *identificação*. Ao verificar os impactos da entrada do LAL RNF *identificação*, identificaram-se os seguintes RNFs específicos que a estratégia influi positivamente: *confidencialidade* e *integridade*. Escolheu-se *confidencialidade* como sendo o RNF específico que a noção em questão pretende satisfazer, pois o acesso às informações mantidas pela interface deve ser concedido somente às lojas autorizadas. A noção em questão foi modificada para indicar a estratégia de satisfação que especializa e o RNF específico que satisfaz: *Possui um código de acesso para utilizar a interface (identificação satisfazendo confidencialidade)*. As entradas do LAL RNF *identificação* e *confidencialidade* passaram a ser entradas do LAL do Udi-RNF. A entrada *Loja* do LAL do Udi-RNF é mostrada na Figura 6.

Para cada impacto da entrada do LAL do Udi, faça:

- 2.2. Verificar se o impacto especializa estratégia de satisfação e satisfaz RNFs, utilizando as entradas do LAL RNF. Caso o impacto especialize uma estratégia de satisfação e satisfaça RNFs, registrar esta informação, entre parêntesis, ao final do impacto.

**Exemplo:** Ao percorrer as entradas do LAL RNF classificadas como estratégia de satisfação, percebeu-se que o impacto *Loja preenche os campos não informados pelo cliente com a sigla 'NE' (não existente)* da entrada do LAL do Udi *Conferir ficha cadastral* (Figura 1) especializava a entrada do LAL RNF *validação* (Figura 5). Ao verificar os impactos da entrada do LAL RNF *validação*, identificaram-se os seguintes RNFs específicos influenciados positivamente pela estratégia: *precisão de valor* e *precisão de propriedade*. Escolheu-se *precisão de propriedade* como sendo o RNF específico que o impacto em questão pretende satisfazer, pois está garantindo a seguinte propriedade: todos os campos da ficha cadastral não informados pelo cliente devem ter a sigla 'NE' (não existente). O impacto sob verificação foi modificado para indicar a estratégia de satisfação que especializa e o RNF específico que satisfaz: *Loja preenche os campos não informados pelo cliente com a sigla 'NE' (não existente) (validação satisfazendo precisão de propriedade)*. As entradas do LAL RNF *validação* e *precisão de propriedade* passaram a ser entradas do LAL do Udi-RNF. A entrada *Conferir ficha cadastral* do LAL do Udi-RNF, construída no decorrer dos exemplos, é mostrada na Figura 7.

3. Identificar novas entradas, noções e impactos, utilizando as entradas do LAL RNF. As novas

Conferir ficha cadastral/Confere ficha cadastral/Conferir contrato
<b>Noções</b>
<ul style="list-style-type: none"> <li>Tarefa realizada pela <i>loja</i> que consiste em verificar se as informações da <i>ficha cadastral</i> estão corretas.</li> <li>Acontece depois que <i>cliente</i> entrega <i>ficha cadastral</i> e apresenta <i>carteira de identidade</i> e <i>CPE</i> para a <i>loja</i>.</li> <li>É uma especialização da estratégia de satisfação <i>validação</i>.</li> <li>Objetiva satisfazer o(s) seguinte(s) RNF(s) primário(s): <i>precisão</i>.</li> <li>Objetiva satisfazer o(s) seguinte(s) RNF(s) específico(s): <i>precisão de valor</i> e <i>precisão de propriedade</i>.</li> </ul>
<b>Impactos</b>
<ul style="list-style-type: none"> <li><i>Loja</i> preenche os campos da <i>ficha cadastral</i> não informados pelo <i>cliente</i> com a sigla 'NE' (não existente) (<i>validação satisfazendo precisão de propriedade</i>).</li> <li><i>Loja</i> verifica o número da <i>identidade</i> presente na <i>ficha cadastral</i> com a <i>carteira de identidade</i> do <i>cliente</i> (<i>validação satisfazendo precisão de valor</i>).</li> <li><i>Loja</i> verifica o número de <i>CPE</i> presente na <i>ficha cadastral</i> com o <i>CPE</i> do <i>cliente</i> (<i>validação satisfazendo precisão de valor</i>).</li> <li><i>Loja</i> verifica o endereço e o telefone de contato presentes na <i>ficha cadastral</i>, ligando para o telefone de contato (<i>validação satisfazendo precisão de valor</i>).</li> <li>Pode entrar em conflito com o(s) seguinte(s) RNF(s) primário(s): <i>precisão</i> e <i>segurança</i>.</li> <li>Pode entrar em conflito com o(s) seguinte(s) RNF(s) específico(s): <i>precisão de oportunidade</i> e <i>confidencialidade</i>.</li> </ul>

Figura 7. Entrada *Conferir ficha cadastral* de um LAL do Udi-RNF



entradas, noções e impactos são verificados nos passos anteriores, quanto a especialização de estratégias de satisfação e satisfação de RNFs.

**Exemplo:** Ao percorrer as entradas do LAL RNF da classe RNF primário, percebeu-se que a entrada do LAL RNF *precisão* tinha relevância para a entrada do LAL do Udi *ficha cadastral* (Figura 1). Ao verificar as noções da entrada do LAL RNF *precisão* (Figura 5), recuperaram-se as seguintes entradas do LAL RNF classificadas como RNF específico: *consistência externa, consistência interna, precisão de oportunidade, precisão de propriedade, precisão de valor e precisão um para um*. Considerou-se que os RNFs específicos *precisão de valor e precisão de propriedade* possuíam relevância para a entrada do LAL do Udi *ficha cadastral*. Ao verificar as noções das entradas do LAL RNF *precisão de valor* (Figura 5) e *precisão de propriedade*, foram identificadas e recuperadas algumas entradas do LAL RNF da classe estratégia de satisfação como, por exemplo: *auditoria de precisão, aviso preventivo de imprecisão, confirmação, faixa de valores, informações de apoio e validação*. Utilizando a entrada do LAL RNF *validação* (Figura 5), identificou-se a entrada do LAL do Udi *Conferir ficha cadastral* (Figura 1) e o impacto *Loja confere ficha cadastral* das entradas do LAL do Udi *Loja e Ficha cadastral* (Figura 1).

### 3.2. Construir cenários

A atividade *construir cenários* da estratégia OORNF é uma extensão do processo de construção de cenários a partir do LAL do Udi, proposto de Hadad [16]. Esta extensão reside basicamente na utilização do LAL do Udi-RNF como entrada para o processo de construção, ao invés do LAL do Udi, e na adição de heurísticas para elicitar restrições de episódios e restrições de recursos. A atividade construir cenários é formada pelos seguintes passos: *identificar atores do Udi, identificar cenários candidatos e descrever cenários candidatos*. O processo de construção de cenários é apresentado, a seguir. Mostra-se um exemplo de construção de cenário a partir do LAL do Udi-RNF de um SFA. As palavras sublinhadas são símbolos do LAL do Udi-RNF.

**Passo 1: Identificar atores do Udi.** Recuperar os símbolos do LAL do Udi-RNF classificados como sujeito, os quais representam os atores do Udi.

**Exemplo:** No LAL do Udi-RNF de um SFA, identificaram-se os seguintes atores do Udi: *central de atendimento, cliente, financeira, interface, ligue-cheque e loja* (Figura 6).

**Passo 2: Identificar cenários candidatos.** Recuperar os impactos das entradas do LAL do Udi-RNF, associadas aos atores do Udi, tendo o cuidado de eliminar os repetidos. Estes impactos são os títulos de cenários candidatos.

**Exemplo:** A partir dos impactos da entrada *loja* do LAL do Udi-RNF (Figura 6), identificaram-se alguns cenários candidatos como, por exemplo: *Loja confere ficha cadastral, Loja aprova crédito e Loja não aprova crédito*.

**Passo 3: Descrever cenários candidatos**

Para cada cenário candidato identificado no passo 2, faça:

Se o título do cenário candidato possui um símbolo do LAL do Udi-RNF classificado como verbo, então:

**Exemplo:** Apresenta-se, neste passo, a descrição do cenário candidato *Loja confere ficha cadastral* (Figura 3).

1. Recuperar a entrada do LAL do Udi-RNF referente ao símbolo classificado como verbo, presente no título do cenário candidato.  
**Exemplo:** Recuperou-se a entrada *Confere ficha cadastral* (Figura 7).
2. Define-se o objetivo do cenário com base no seu título e nas noções da entrada do LAL do Udi-RNF, recuperada no item 1 acima.  
**Exemplo:** Ao verificar a primeira noção da entrada *confere ficha cadastral* (Figura 7), obteve-se o objetivo do cenário, que é *verificar se as informações da ficha cadastral estão corretas*.
3. Define-se o contexto do cenário com base nas noções da entrada do LAL do Udi-RNF, recuperada no item 1 acima. Verifica-se também se existe alguma ordem de precedência entre o impacto que originou o cenário e os outros impactos. Caso exista, inclui-se, no contexto do cenário, o impacto que o precede como uma pré-condição.  
**Exemplo:** Ao verificar a segunda noção da entrada *confere ficha cadastral*, obtém-se uma pré-condição do cenário, a qual é colocada em seu contexto: cliente entrega ficha cadastral e apresenta carteira de identidade e CPF para a loja.
4. A partir dos impactos da entrada do LAL do Udi-RNF, recuperada no item 1 acima, excetuando-se aqueles que indicam possíveis conflitos da entrada com RNFs, obtém-se os episódios do cenário. Se o impacto, a partir do qual o episódio foi definido, especializa uma estratégia de satisfação e satisfaz RNFs específicos, o episódio também especializa a mesma estratégia de satisfação e satisfaz os mesmos RNFs específicos. Neste caso, deve-se incluir uma restrição para o episódio, seguindo o formato: **Restrição:** *1{Recurso}N deve(m) ter 1{RNFespecifico}n, sendo a estratégia de satisfação estratégia de satisfação que o episódio especializa*.  
**Exemplo:** A partir do impacto *Loja preenche os campos da ficha cadastral não informados pelo cliente com a sigla 'NE' (não existente) (validação satisfazendo precisão de propriedade)* da entrada *confere ficha cadastral* (Figura 7), define-se o seguinte episódio do cenário *Loja confere ficha cadastral*: *Loja preenche os campos da ficha cadastral não informados pelo cliente com a sigla 'NE' (não existente)*. Como o impacto especializa a estratégia de satisfação *validação* e satisfaz o RNF específico *precisão de propriedade*, inclui-se a seguinte restrição ao episódio: **Restrição:** *ficha cadastral deve ter precisão de propriedade, sendo a estratégia de satisfação validação*.
5. Os atores do cenário são entradas do LAL do Udi-RNF presentes em episódios, classificadas como sujeito e que executam ações no cenário.  
**Exemplo:** Ao verificar os episódios do cenário *Loja confere ficha cadastral* (Figura 3), identificou-se o ator *loja*.
6. Os recursos são entradas do LAL do Udi-RNF, presentes nos episódios do cenário e classificadas como objeto. Deve-se também examinar as restrições de episódios para identificar restrições de recursos, as quais devem obedecer ao seguinte formato: **Restrição:** *deve ter 1{RNFespecifico}n*.  
**Exemplo:** Ao verificar os episódios do cenário *Loja confere ficha cadastral* (Figura 3), identificaram-se os recursos *ficha cadastral, carteira de identidade e CPF*. Ao examinar as restrições dos episódios, definiu-se a seguinte restrição para o recurso *ficha cadastral*: **Restrição:** *deve ter precisão de propriedade e precisão de valor*.

### 3.3. Construir diagrama de classes

A atividade *construir diagrama de classes* da estratégia OORNF é uma extensão da estratégia de derivação proposta por Leite [18] e Leonardi [19]. Esta extensão consiste basicamente na utilização do LAL do Udi-RNF e dos cenários, construídos nas atividades anteriores da estratégia OORNF, como entradas para o processo e na modificação dos passos da estratégia original para que considerem as informações sobre RNFs e estratégias de satisfação, presentes no LAL do Udi-RNF e nos cenários. A atividade construir diagrama de classes é constituída pelos seguintes passos: *identificar classes primárias e suas responsabilidades*, *identificar classes secundárias e suas responsabilidades*, *refinar responsabilidades e colaborações* e *construir diagrama de classes conceitual*. Classes primárias são entidades ativas do Udi como, por exemplo, pessoas e organizações. Classes secundárias são entidades passivas do Udi (em geral, são repositórios de dados). Descreve-se abaixo o processo de construção de diagramas de classes. Mostram-se alguns exemplos, utilizando-se, como entrada para o processo, o LAL do Udi-RNF e os cenários de um SFA. As palavras sublinhadas são símbolos do LAL do Udi-RNF.

#### Passo 1: Identificar classes primárias e suas responsabilidades.

1. Cada ator dos cenários será uma classe primária.
2. Eliminar classes redundantes.
3. As responsabilidades de cada classe primária são definidas a partir dos impactos da entrada do LAL do Udi-RNF, associada à classe primária. Se o impacto especializa uma estratégia de satisfação e satisfaz RNFs específicos, trazer esta informação juntamente com a responsabilidade.

**Exemplo:** Ao verificar o cenário *Loja confere ficha cadastral* (Figura 3), identificou-se a classe primária *loja*. Ao verificar a entrada *loja* do LAL do Udi-RNF (Figura 6), definiram-se as seguintes responsabilidades: *aprovar crédito*, *conferir contrato (validação satisfazendo precisão de valor e precisão de propriedade)*, *enviar movimento para financeira*, *não aprovar crédito* e *receber código de acesso e senha da financeira*. A responsabilidade *conferir contrato* especializa a estratégia de satisfação *validação* e satisfaz os RNFs específicos *precisão de valor* e *precisão de propriedade*.

#### Passo 2: Identificar classes secundárias e suas responsabilidades.

1. Cada recurso dos cenários será uma classe secundária.
2. Cada entrada do LAL do Udi-RNF, classificada como objeto e presente nas responsabilidades das classes primárias, será uma classe secundária.
3. A união das classes secundárias, identificadas nos itens 1 e 2, contém todas as classes secundárias identificadas.
4. Eliminar classes não significativas, classes que são atributos de outras classes e classes redundantes.
5. As responsabilidades de classes secundárias são geralmente do tipo *registrar informações e disponibilizar informações*. Estas informações são identificadas nas noções das entradas do LAL do Udi-RNF, associadas às classes secundárias.

**Exemplo:** Ao verificar o cenário *Loja confere ficha cadastral* (Figura 3), identificaram-se as seguintes classes secundárias: *carteira de identidade*, *CPF* e *ficha cadastral*.

Ao verificar as noções da entrada *carteira de identidade* do LAL do Udi-RNF (Figura 8), percebeu-se que esta possui as seguintes informações: *número*, *nome*, *filiação* e *órgão expedidor*. Assim, as responsabilidades desta classe são: *registrar número*, *nome*, *filiação* e *órgão expedidor* e *disponibilizar número*, *nome*, *filiação* e *órgão expedidor*.

Carteira de identidade/identidade
<p>Noções</p> <ul style="list-style-type: none"> <li>• Carteira de identidade original do cliente</li> <li>• Possui número, nome, filiação e órgão expedidor.</li> <li>• É uma especialização da estratégia de satisfação <i>informações de apoio</i></li> <li>• Objetiva satisfazer o(s) seguinte(s) RNF(s) primários: <i>precisão</i></li> <li>• Objetiva satisfazer o(s) seguinte(s) RNF(s) específicos: <i>precisão de valor</i></li> </ul> <p>Impactos</p> <ul style="list-style-type: none"> <li>• Cliente apresenta <i>carteira de identidade</i> para o <i>loja</i></li> <li>• Pode entrar em conflito com o(s) seguinte(s) RNF(s) primários: <i>estado e performance</i></li> <li>• Pode entrar em conflito com o(s) seguinte(s) RNF(s) específicos: <i>estado operacional e performance de acesso</i></li> </ul>

Figura 8. Entrada *Carteira de identidade* de um LAL do Udi-RNF

#### Passo 3: Refinar responsabilidades e colaborações.

1. Construir um cartão CRC para cada classe.
2. Verificar se a entrada do LAL do Udi-RNF, associada à classe, indica que esta especializa uma estratégia de satisfação e satisfaz RNFs específicos. Se for o caso, incluir uma justificativa, na parte *justificativa* do cartão CRC, obedecendo ao seguinte formato: *É uma especialização da estratégia de satisfação estratégia de satisfação que a classe especializa e satisfaz o(s) RNF(s) específico(s): 1{RNF\_Específico}N*.
3. Colocar as responsabilidades das classes, identificadas nos passos 1 e 2, na parte *responsabilidades* dos cartões CRC.
4. Verificar as noções da entrada do LAL do Udi-RNF, associada à classe, para identificar relacionamentos de generalização, especialização e agregação. Em seguida, registrar estes relacionamentos nas partes *superclasse*, *subclasses* e *partes* do cartão CRC.
5. Caso seja uma classe primária, identificar responsabilidades adicionais a partir dos episódios dos cenários nos quais a classe participa. Se o episódio, associado à nova responsabilidade adicionada, possui restrição, então incluir as informações sobre a estratégia de satisfação que o episódio especializa e os RNFs específicos que satisfaz, presentes na restrição, no final do texto da responsabilidade, seguindo o formato: *(estratégia de satisfação que a responsabilidade especializa satisfazendo 1{RNF\_Específico}N)*.
6. Para cada responsabilidade de uma classe, verificar com quais outras classes a classe precisa colaborar para cumprir a responsabilidade. Colocar os nomes destas classes na parte *colaborações* do cartão CRC.

**Exemplo 1:** Construiu-se um cartão CRC para a classe primária *loja* (Figura 9). As responsabilidades, identificadas no passo 1, foram colocadas na parte

loja	Colaborações
<p><b>Responsabilidades</b></p> <ul style="list-style-type: none"> <li>• <i>conferir contrato (validação satisfazendo precisão de valor e precisão de propriedade)</i></li> <li>• <i>aprovar crédito</i></li> <li>• <i>não aprovar crédito</i></li> <li>• <i>enviar movimento para financeira</i></li> <li>• <i>receber código de acesso e senha da financeira</i></li> <li>• <i>informar CPF do cliente para ligar-checke</i></li> <li>• <i>solicitar código de aprovação para controle de atendimento</i></li> <li>• <i>anotar no número de contrato do contrato o código de aprovação fornecido pelo control de atendimento</i></li> <li>• <i>anotar número de presente nos clientes</i></li> <li>• <i>preencher os campos do contrato não informados pelo cliente com a sigla NE (não existente) (validação satisfazendo precisão de propriedade)</i></li> <li>• <i>verificar o número da identidade presente no contrato com a carteira de identidade do cliente (validação satisfazendo precisão de valor)</i></li> <li>• <i>verificar o número de CPF presente no contrato com o CPF do cliente (validação satisfazendo precisão de valor)</i></li> <li>• <i>verificar o endereço e o telefone de contato presentes no contrato, ligando para o telefone de contato (validação satisfazendo precisão de valor)</i></li> <li>• <i>fornecer código de acesso para a interface (identificação satisfazendo confiabilidade)</i></li> <li>• <i>fornecer senha para a interface (autenticação satisfazendo confiabilidade)</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>atenda</i></li> <li>• <i>carteira de identidade</i></li> <li>• <i>control de atendimento</i></li> <li>• <i>cliente</i></li> <li>• <i>contrato</i></li> <li>• <i>CPF</i></li> <li>• <i>financeira</i></li> <li>• <i>interface</i></li> <li>• <i>ligar-checke</i></li> <li>• <i>movimento</i></li> </ul>

Figura 9. Cartão CRC loja

responsabilidades do cartão CRC. A partir do episódio *Loja verifica o número da identidade presente na ficha cadastral com a carteira de identidade do cliente* do cenário *Loja confere ficha cadastral*, o qual apresenta a restrição **Restrição: ficha cadastral deve ter precisão de valor, sendo a estratégia de satisfação validação**, foi definida a responsabilidade *verificar o número da identidade presente no contrato com a carteira de identidade do cliente (validação satisfazendo precisão de valor)*. Para cumprir a responsabilidade *conferir contrato*, a classe *loja* precisa colaborar com a classe *contrato*. Para cumprir a responsabilidade *enviar movimento para financeira*, a classe *loja* necessita da colaboração das classes *movimento* e *interface*.

**Exemplo 2:** Construiu-se um cartão CRC para a classe *carteira de identidade* (Figura 10). Ao verificar a entrada *carteira de identidade* do LAL do Udi-RNF (Figura 8), percebeu-se que esta especializa a estratégia de satisfação *informações de apoio* e satisfazia o RNF específico *precisão de valor*. Assim, incluiu-se a seguinte justificativa para o cartão CRC: *É uma especialização da estratégia de satisfação informações de apoio e satisfaz o(s) RNF(s) específico(s) precisão de valor*. As responsabilidades, identificadas no passo 2, foram colocadas na parte *responsabilidades* do cartão CRC. Não foram identificadas colaborações.

carteira de identidade	
Justificativa: É uma especialização da estratégia de satisfação informações de apoio e satisfaz o(s) RNF(s) específico(s) precisão de valor	
Responsabilidades	Colaborações
<ul style="list-style-type: none"> <li>• registrar número, nome, filiação e órgão expedidor,</li> <li>• disponibilizar número, nome, filiação e órgão</li> </ul>	

Figura 10. Cartão CRC carteira de identidade

**Passo 4: Construir diagrama de classes conceitual.**

1. Identificar atributos e operações

1.1. Para cada cartão CRC, construir uma classe seguindo qualquer notação orientada a objetos. Se o cartão CRC especializa uma estratégia de satisfação e satisfaz RNFs específicos, incluir o prefixo RN\_ no nome da classe.

1.2. Examinar as noções da entrada do LAL do Udi-RNF e as responsabilidades do cartão CRC, referentes à classe, para identificar atributos da classe. Se a noção/responsabilidade especializa uma estratégia de satisfação e satisfaz RNFs específicos, incluir o prefixo RN\_ no nome do atributo identificado a partir da noção/responsabilidade.

1.3. A partir das responsabilidades, presentes no cartão CRC, definem-se as operações da classe. Se a responsabilidade especializa uma estratégia de satisfação e satisfaz RNFs específicos, incluir o prefixo RN\_ no nome da operação.

**Exemplo 1:** Construiu-se uma classe (Figura 11) para o cartão CRC *loja* (Figura 9), seguindo a linguagem UML[20]. Ao examinar as noções da entrada *loja* do LAL do Udi-RNF (Figura 6), identificaram-se os atributos *código de acesso* e *senha*. Como as noções, a partir das quais estes atributos foram identificados, especializavam respectivamente as estratégias de satisfação *identificação* e *autenticação* e satisfaziam o RNF específico *confidencialidade*, incluiu-se o prefixo

Loja	
RN_Código de acesso	
RN_Senha	
RN_ConferirContrato(contrato)	
AprovarCredito(contrato)	
NãoAprovarCredito(contrato)	
EnviarMovimentoParaFinanceira(movimento)	
ReceberCodigoSenhaDaFinanceira(codigo de acesso, senha)	
InformarCPPParaLiquarCheque(contrato, liquar-cheque)	
SolicitarCodigoDeAprovacaoParaCentralDeAtendimento()	
AnotarCodigoDeAprovacaoNoNumeroDeContrato(contrato, codigo de aprovacao)	
AnotarNumeroDePropostaNosCheques(contrato, cheques)	
RN_PreencherCamposNaoInformadosComNI(contrato)	
RN_VerificarNumeroDeIdentidadeCarteiraDeIdentidade(contrato)	
RN_VerificarNumeroDeCPF(contrato)	
RN_VerificarTelefoneEnderco(contrato)	
RN_FornecerCodigoDeAcesso()	
RN_FornecerSenha()	

Figura 11. Classe loja

RN\_ nos nomes destes atributos. A operação *RN\_ConferirContrato(contrato)* foi definida a partir da responsabilidade *conferir contrato (validação satisfazendo precisão de valor e precisão de propriedade)*. Incluiu-se o prefixo RN\_ no nome da operação para indicar que esta especializa uma estratégia de satisfação e satisfaz RNFs específicos.

**Exemplo 2:** Construiu-se uma classe (Figura 12) para o cartão CRC *carteira de identidade* (Figura 10), seguindo a linguagem UML[20]. Como o cartão CRC *carteira de identidade* especializa a estratégia de satisfação *informações de apoio* e satisfaz o RNF específico *precisão de valor*, incluiu-se o prefixo RN\_ no nome da classe. Ao examinar as responsabilidades do cartão CRC *carteira de identidade*, identificaram-se os seguintes atributos: número, nome, filiação e órgão expedidor. Definiram-se as operações da classe em questão a partir das responsabilidades do cartão CRC *carteira de identidade*.

RN_Carteira de identidade	
Número	
Nome	
Órgão expedidor	
Filiação	
RegistrarNúmero(numero)	
DisponibilizarNúmero()	
RegistrarNome(nome)	
DisponibilizarNome()	
RegistrarFiliação(filiação)	
DisponibilizarFiliação()	
RegistrarOrgãoExpedidor(orgão expedidor)	
DisponibilizarOrgãoExpedidor()	

Figura 12. Classe carteira de identidade

2. Identificar relacionamentos

2.1. Para cada cartão CRC, construir novamente uma classe, seguindo qualquer notação orientada a objetos e sem mostrar atributos e operações. Se o cartão CRC especializa uma estratégia de satisfação e satisfaz RNFs específicos, incluir o prefixo RN\_ no nome da classe.

2.2. Para cada classe primária, identificar as colaborações desta com outras classes, no cartão CRC correspondente. Representar associações para cada colaboração identificada. Os nomes

destas associações são obtidos nas responsabilidades do cartão CRC ou nas noções da entrada do LAL do Udi-RNF, associados à classe primária. Se a responsabilidade ou noção especializa uma estratégia de satisfação e satisfaz RNFs específicos, acrescentar o prefixo RN\_ no nome da associação. A multiplicidade das associações pode ser obtida nas noções da entrada do LAL do Udi-RNF ou nas responsabilidades do cartão CRC, referentes à classe primária. Pode-se incluir mais de uma associação para uma colaboração, caso isto contribua para aumentar a compreensão do Udi.

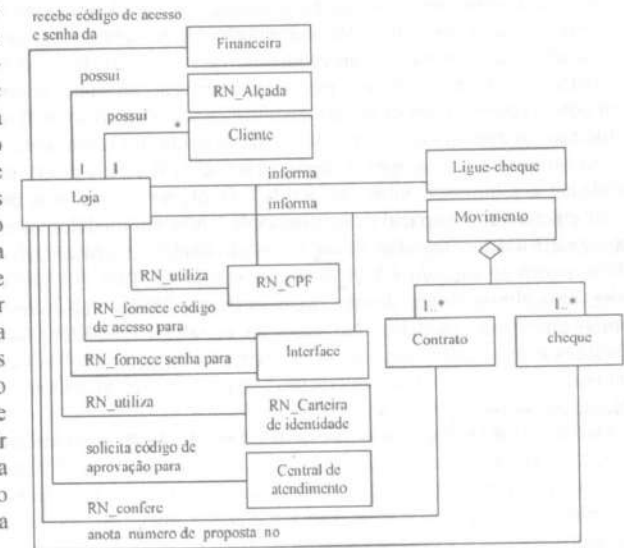


Figura 13. Parte do diagrama de classes de um SFA

2.3. Os relacionamentos de generalização, especialização e agregação são construídos a partir das informações dos cartões CRC, registradas nas partes *superclasse*, *subclasses* e *partes*.

**Exemplo 1:** A partir dos cartões CRC do SFA, representaram-se classes, utilizando a linguagem UML [20] como, por exemplo: *RN\_alçada*, *RN\_carteira de identidade*, *cliente*, *contrato*, *financeira*, *loja* e *movimento*. O prefixo *RN\_* foi acrescentado aos nomes das classes que especializam estratégias de satisfação e satisfazem RNFs específicos.

**Exemplo 2:** Para a classe primária *loja*, identificaram-se, no cartão CRC *loja* (Figura 9), colaborações com outras classes como, por exemplo: *RN\_alçada* e *contrato*. Para a colaboração com *contrato*, incluiu-se a associação *RN\_Confere*, sendo o nome da associação identificado na responsabilidade *conferir ficha cadastral (validação satisfazendo precisão de valor e precisão de propriedade)*. Acrescentou-se o prefixo *RN\_* ao nome da associação, pois esta especializa a estratégia de satisfação *validação* e satisfaz os RNFs específicos *precisão de valor* e *precisão de propriedade*. Para a colaboração com *RN\_Alçada*, incluiu-se a associação *possui*. O nome desta associação foi obtido a partir da noção *Possui uma alçada, a qual é definida pela financeira* da entrada *Loja* do LAL do Udi-RNF (Figura 6). A parte do diagrama de classes, construído para o SFA a partir da classe primária *loja*, é mostrada na Figura 13.

#### 4. Trabalhos relacionados

Chung propõe um framework (*NFR Framework*) para representar e usar RNFs no processo de desenvolvimento de sistemas [9]. O *NFR framework* é composto por 5 elementos básicos: um conjunto de metas para representar RNFs, decisões de desenho (também chamadas de estratégias de satisfação) e argumentos; um conjunto de tipos de ligação para relacionar metas com outras metas e com ligações; um conjunto de métodos genéricos para refinar metas e ligações em outras metas e ligações; um conjunto de regras de correlação para inferir relações potenciais entre metas e um procedimento rotulador para determinar o grau de satisfação de RNFs através das decisões de desenho. As metas e ligações são representadas através de grafos (grafos de RNFs). A partir do trabalho em questão, compreendemos que RNFs podem ser decompostos em RNFs mais detalhados e satisfeitos por estratégias de satisfação, as quais podem também entrar em conflito com outros RNFs. Além disso, utilizamos o conhecimento sobre RNFs e estratégias de satisfação, registrado no trabalho de Chung, para criar as primeiras entradas do LAL RNF.

Cysneiros e Leite propõem uma estratégia (ER-RNF) para integrar RNFs aos modelos de entidades e relacionamentos (MER)[21]. O objetivo é tratar o problema de representação de RNFs e entender o impacto da satisfação de RNFs em modelos conceituais de banco de dados. A estratégia ER-RNF engloba as seguintes atividades: *construir LAL do Udi*, *construir grafos de RNFs*, *construir modelo ER* e *construir modelo ER-RNF*. A partir do trabalho de Cysneiros e Leite, abstraímos o fato de que, para satisfazer RNFs, faz-se necessário a inclusão de novos componentes nos modelos. No caso da estratégia ER-RNF, são incluídos novos atributos, entidades e relacionamentos no MER para satisfazer RNFs. No caso da estratégia OORNF, são incluídos novos atributos, operações, classes e relacionamentos no diagrama de classes, objetivando satisfazer RNFs.

Macedo e Leite propõem uma estratégia de elicitação e representação de RNFs e de requisitos baseados em ações concretas [6]. Para representar os RNFs juntamente com os requisitos baseados em ações concretas, realiza-se uma extensão ao modelo básico do SERBAC (Suporte à Engenharia de Requisitos com Base em Ações Concretas). Para elicitar RNFs em conjunto com requisitos baseados em ações concretas, são utilizados o *acesso estruturado* e o *repositório de*

*conhecimento de RNFs de qualidade*. Em nosso trabalho, estendemos o repositório de conhecimento de RNFs. Esta extensão consiste basicamente no registro de informações adicionais como, por exemplo, descrições de RNFs e estratégias de satisfação, conflitos entre RNFs, estratégias de satisfação que satisfazem RNFs e influências positivas e negativas de estratégias de satisfação sobre RNFs específicos. Além disso, apresentamos esta base de conhecimento como um Léxico Ampliado da Linguagem de RNFs (LAL RNF), o qual é uma representação em linguagem natural que facilita a assimilação e utilização do conhecimento sobre RNFs por parte de engenheiros de requisitos/engenheiros de software.

Os trabalhos apresentados nesta seção não consideram RNFs no contexto da orientação a objetos. Portanto, acreditamos não ser adequado compará-los à estratégia proposta neste trabalho.

#### 5. Conclusão

Visto que a maioria dos métodos atuais da engenharia de software e da engenharia de requisitos vêm negligenciando a consideração de RNFs, a principal contribuição deste trabalho reside no suporte à elicitação e à modelagem de RNFs e da satisfação de RNFs, durante a construção de LAL do Udi, cenários, cartões CRC e diagramas de classes. Além disso, apesar da completude de requisitos ser uma impossibilidade [1], os modelos construídos com a estratégia proposta são mais completos, por registrarem RNFs e satisfação de RNFs, quando comparados com modelos produzidos através de processos e métodos que não consideram RNFs.

O LAL RNF é uma importante técnica de elicitação de RNFs e estratégias de satisfação, pois registra conhecimento sobre RNFs e estratégias de satisfação, o qual é utilizado para verificar quais RNFs e estratégias de satisfação são relevantes no Udi em questão. Além disso, ao usar o LAL RNF, identificam-se entradas do LAL do Udi que dificilmente seriam elicítadas sem a utilização deste. Essas novas entradas resultam, ao aplicar a estratégia proposta, em novos cenários, classes primárias, classes secundárias, responsabilidades, atributos e operações.

Neste trabalho, com a identificação e registro de especializações de estratégias de satisfação e da conseqüente satisfação de RNFs, obtém-se uma justificação e explicação dos componentes dos modelos através de RNFs. Assim, consegue-se justificar e explicar, por exemplo, com base em RNFs, a presença de classes, atributos, operações e relacionamentos de diagramas de classes.

Aplicamos a estratégia proposta em três estudos de caso detalhados (sistema de informações laboratoriais, sistema de financiamento com alçada e sistema de informação da Pós-graduação) [22]. Em todos os casos, foram elicítados e modelados RNFs e satisfação de RNFs que ainda não tinham sido elicítados e modelados anteriormente. Além disso, desenvolvemos uma ferramenta que apóia a aplicação de parte da estratégia proposta (até a construção de cartões CRC) [22]. Vale ressaltar também que alguns passos da estratégia são semi-automatizados pela ferramenta como, por exemplo, a construção de cenários a partir do LAL do Udi-RNF e a construção de cartões CRC a partir do LAL do Udi-RNF e dos cenários.

Futuramente, pretendemos investigar a evolução do LAL do Udi-RNF, os conseqüentes impactos desta evolução nos modelos construídos a partir do LAL do Udi-RNF e estratégias de gerência desta evolução. Objetivamos também realizar extensões ao Rational Unified Process (RUP) [20] e à Unified Modeling Language (UML) [20] a fim de fornecer suporte à elicitação e modelagem de RNFs e satisfação de RNFs.

#### 7. Bibliografia

- [1] Leite, J.C.S.P. *Engenharia de Requisitos*. Departamento de Informática, PUC-Rio, 1994.



- [2] McCall, J.A. et. al. *Factors in Software Quality*. Vol. 1, 2 e 3, AD/A-049-014/015/055, Nat'l Tech. Information Service, Springfield, Va., 1977.
- [3] Boehm, B. *Characteristics of Software Quality*. North Holland Press, 1978.
- [4] ISO 9126 *Information Technology – Software Product Evaluation – Quality Characteristics and Guidelines for their Use*, International Organization for Standardization, Geneva, 1992.
- [5] Yeh, R. et. al. *Software Requirements: New Directions and Perspectives*. Handbook of Software Engineering, 1984, pp. 519-543.
- [6] Macedo, N.A.M. e Leite, J.C.S.P. *Integrando Requisitos Não Funcionais aos Requisitos Baseados em Ações Concretas*. 2º Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes Software (IDEAS'99), San José, Costa Rica, 1999.
- [7] Roman, G.-C. *A Taxonomy of Current Issues in Requirements Engineering*. IEEE Computer, 18(4), 1985, pp.14-22.
- [8] Brackett, J.W. *Software Requirements*. SEI-CM-19-1.2, Software Engineering Institute, January 1990.
- [9] Chung, K.L. *Representing and Using Non-functional Requirements: A Process-oriented approach*. Ph.D. Thesis, Department of Computer Science, University of Toronto, 1993.
- [10] Landes, D. and Studer, R.. *The Treatment of Non-Functional Requirements in MIKE*. LNCS 989 (Software Engineering:ESEC'95), Springer-Verlag, 1995, pp. 294-306
- [11] Ebert, C.. *Dealing with Nonfunctional in Large Software Systems*. Annals of Software Engineering, 3, 1997, pp. 367-395.
- [12] Kotonya, G. and Sommerville, I. *Requirements Engineering: Processes and Techniques*. John Willey & Sons, 1998.
- [13] Breitman, K. K., Leite J.C.S.P. and Finkelstein A. *The World's Stage: A Survey on Requirements Engineering Using a Real-Life Case Study*. Journal of the Brazilian Computer Society, vol. 6, n.1, July 1999.
- [14] Monarchi, D. et.al. *A Research Typology for Object-Oriented Analysis and Design*. Communications of the ACM, 35, September 1992, pp. 35-47.
- [15] Franco, A. P. M. *Métodos e Representação de Suporte à Aquisição de Linguagens de Aplicação*. Dissertação de Mestrado, Departamento de Informática, PUC-Rio, 1992.
- [16] Hadad, G. et. al. *Construcción de Escenarios a partir del Léxico Extendido del Lenguaje*. JAIIO'97, Argentina, 1997, pp. 65-77
- [17] Leite, J.C.S.P. et.al. *Enhancing a Requirements Baseline with Scenarios*. Requirements Engineering, 2(4):184-198, 1997.
- [18] Leite, J.C.S.P. e Breitman, K. K. Tutorial: Utilizando Cenários para o Desenvolvimento de Sistemas Orientados a Objetos. Simpósio Brasileiro de Engenharia de Software, Brasil, 1997.
- [19] Leonardi, C. et. al. *Una Estrategia de Análisis Orientada a Objetos basada en Escenarios*. Actas II Jornadas de Ingeniería de Software JIS97, España, Set. 1997.
- [20] *UML Resource Center*.. <http://www.rational.com/uml>
- [21] Cysneiros, L.M. and Leite, J.C.S.P. *Integrating Non-Functional Requirements into Data Modeling*. Fourth IEEE International Symposium on Requirements Engineering, 1999.
- [22] Sabát Neto, J. de M. *Integrando Requisitos Não Funcionais à Modelagem Orientada a Objetos*. Dissertação de Mestrado, Departamento de Informática, PUC-Rio, 2000.