



PUC

ISSN 0103-9741

Monografias em Ciência da Computação
nº 24/09

Validação de Cenários

Fillipe Machado Pinto Napolitano
Simone Diniz Junqueira Barbosa
Julio Cesar Sampaio do Prado Leite

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO
RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22451-900
RIO DE JANEIRO - BRASIL

Validação de Cenários

Fillipe Machado Pinto Napolitano, Simone Diniz Junqueira Barbosa,
Julio Cesar Sampaio do Prado Leite

fnapolitano@inf.puc-rio.br, simone@inf.puc-rio.br, www.inf.puc-rio.br/~julio

Abstract. Scenario is a description technique which has become more evident in many projects. For using natural language in the communication between requirements engineers and stakeholders, scenarios have contributed directly to the collaboration among all participants in the requirements definition process. The main contribution of this work is the elaboration of a simulation-based scenario validation strategy. This strategy makes use of interactive techniques, allowing for better adjustments in the elicited information by the requirements engineer.

Keywords: Scenarios, Requirements Elicitation, Validation, Simulation, Softgoals.

Resumo. Cenário é uma técnica de descrição que vem se tornando bastante evidente em diversos projetos. Por utilizar linguagem natural na comunicação entre engenheiros de requisitos e interessados, os cenários têm contribuído diretamente na colaboração entre todos os participantes do processo de definição de requisitos. Este trabalho tem como principal contribuição à elaboração de uma estratégia baseada em simulação para a validação de cenários. Essa estratégia faz uso de técnicas bastante interativas, promovendo assim um melhor ajuste no entendimento das informações elicítadas pelos engenheiros de requisitos.

Palavras-chave: Cenários, Elicitação de Requisitos, Validação, Simulação, Metas Flexíveis.

Responsável por publicações:

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC-Rio Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22451-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3527-1516 Fax: +55 21 3527-1530
E-mail: bib-di@inf.puc-rio.br
Web site: <http://bib-di.inf.puc-rio.br/techreports/>

Sumário

1	Introdução	1
2	Trabalhos Relacionados	1
3	Visão Geral da Estratégia	2
4	Detalhamento da Estratégia	3
4.1	CONSTRUIR CENÁRIOS	4
4.2	TRANSFORMAR	6
4.2.1	Tratando Restrições do Tipo Metas Flexíveis	8
4.3	VALIDAR POR SIMULAÇÃO	11
5	Um Estudo Preliminar da Estratégia	13
6	Conclusões	16
	Referências	17
	Apêndice 1 Exemplo de Formulário de Validação	19
	Apêndice 2 Cenários Sistema de Controle de Caixa de Restaurante	21
	Apêndice 3 Cenários Transformados para a Ferramenta	23
	Apêndice 4 Léxico Sistema Controle do Caixa de Restaurante	25

1 Introdução

A necessidade de garantir um bom entendimento entre engenheiros de requisitos e interessados tem motivado pesquisadores no desenvolvimento de métodos e técnicas que permitam melhores resultados na colaboração entre todos os participantes no processo de definição de requisitos [1].

Cenário é uma técnica bem conhecida que ajuda no melhor entendimento dos requisitos de um sistema. São construídos de diversas formas na literatura, indo da forma narrativa até a de protótipos [14]. Neste trabalho, será utilizada a abordagem apresentada em [6], onde cenários descrevem situações que ocorrem no macrosistema e suas relações com o sistema, evoluem durante o processo de desenvolvimento, são naturalmente ligados ao léxico e são descritos em linguagem natural [4].

Diversas disciplinas vêm utilizando cenários atualmente como meio de melhor entender o domínio da aplicação, tornar mais eficiente a comunicação e o comprometimento com os interessados e melhorar a organização das informações envolvidas.

Este trabalho detalha uma estratégia baseada em simulação para validação de cenários. Tal fato está relacionado ao problema gerencial de qualidade dos cenários, ou seja, da confiabilidade das informações descritas nos mesmos. Espera-se que a estratégia detalhada aqui possa contribuir para o aumento de conhecimento em gerenciamento de cenários.

Essa monografia está dividida em seções, organizadas da seguinte forma: na seção 2 são descritos os trabalhos relacionados; na seção 3 é apresentada a visão geral da estratégia; na seção 4 a estratégia é detalhada; na seção 5 é apresentado um estudo preliminar da estratégia e, finalmente, na seção 6 são documentadas as considerações finais e os futuros desdobramentos da pesquisa proposta.

2 Trabalhos Relacionados

O trabalho documentado nessa monografia foi influenciado por vários trabalhos relacionados à construção, verificação e validação de cenários. Alguns deles são apresentados a seguir:

Em [1] é apresentado um processo para construção de cenários. Este trabalho mostra algumas diferentes abordagens para construção de cenários e em seguida, apresenta seu próprio processo. O processo proposto é constituído por cinco atividades: Produzir, Descrever, Organizar, Verificar e Validar. Leite et al. afirma que a construção de cenários baseia-se inicialmente no léxico do domínio da aplicação, produzindo uma versão inicial de cenários. Esses cenários são melhorados usando outras fontes de informação e são organizados, visando obter um conjunto consistente, representando o domínio da aplicação. Durante ou após essas atividades, os cenários são verificados e validados, sendo detectadas assim, discrepâncias, erros e/ou omissões (DEO). É importante ressaltar que este trabalho cuida de importantes problemas relativos ao gerenciamento de cenários, com foco na organização dos mesmos.

Em [2], mostra-se como inspeções podem ajudar aos desenvolvedores de software a gerenciar melhor a produção de cenários. Usando as inspeções propostas por Fagan [3] como principal paradigma, este trabalho propõe uma estratégia de inspeção baseada em detecção de defeitos por formulários. Partindo do princípio que a flexibilidade e

simplicidade dos cenários permitem a existência de defeitos, como falta/erro de informação, ambiguidades, contradições e sobreposições, Leite et al. apresenta uma taxonomia de defeitos, dividindo os mesmos em 3 grupos: *Discrepancies*, *Errors* e *Omissions* (DEO). Em seguida, [2] mostra as atividades do processo de inspeção proposto (Planejar, Preparar, Encontrar e Retrabalhar) e como os formulários propostos para detecção de defeitos (intracenários e intercenários) são utilizados na melhoria da qualidade dos cenários construídos. Por fim, um outro critério de classificação é apresentado, atribuindo a cada defeito, também, um grau de severidade (*Fundamental*, *Organizational* ou *Presentation*).

Em [19] é apresentado um estudo de caso que examina a elicitación e validación de cenários durante o projeto de um sistema para máquinas de radioterapia. O sistema tem a finalidade de controlar, monitorar e prover uma interface de operação e serviços para os equipamentos. Neste estudo, os cenários foram introduzidos para permitir uma melhor interação dos interessados durante o processo de requisitos. Para a validación dos cenários produzidos, foi utilizado o método formal de inspeções de Fagnan [3]. Os resultados finais foram satisfatórios, porém alguns tópicos da pesquisa poderiam ser melhorados, como a compreensibilidade e estrutura do documento de cenários e a visão da interação entre os mesmos. Este fato levou os pesquisadores a adotar uma representação gráfica mais expressiva, como multimídia e diagramas de fluxo, em um software chamado *Scenario Manager*. Experiências iniciais mostraram grande potencial no emprego das técnicas supracitadas na compreensibilidade dos cenários, melhorando assim a qualidade da documentação de requisitos.

O trabalho [20] referencia dois difíceis problemas da engenharia de requisitos: previsibilidade e rastreabilidade. Haumer et al. trata esses problemas de maneira combinada, através de um *framework* arquitetural desenvolvido no projeto europeu CREWS (*Cooperative Requirements Engineering with Scenarios*). É apresentada também a metodologia, implementação e avaliação preliminar deste *framework*, chamado CREWS-EVE. O CREWS-EVE é um ambiente que integra multimídia, cenários e animação para a solução dos problemas propostos, em contrapartida a utilização de cenários baseados em texto, devido principalmente as suas limitações. Um estudo de caso mostrou que a integração das soluções para os problemas de previsibilidade e rastreabilidade aliadas ao ambiente CREWS-EVE proveu benefícios importantes que suavizaram as limitações encontradas na aplicação das técnicas separadamente.

3 Visão Geral da Estratégia

A estratégia proposta tem por objetivo ajudar ao engenheiro de requisitos a validar os cenários junto aos interessados, utilizando para isso, técnicas interativas para ajustar seu modelo proposto do domínio do problema e corrigir falhas apontadas pelos interessados durante o processo de validação. Essas técnicas interativas envolvem simulação [7],[8],[9], formulários [1],[10] e registro de impressões em vídeo durante a realização da simulação [10]. O modelo SADT¹ [5] a seguir descreve a visão geral da estratégia.

1 Notação do SADT: caixas representam atividades, setas a esquerda representam as entradas das atividades, setas a direita as saídas, setas acima representam mecanismos e setas abaixo representam controles.

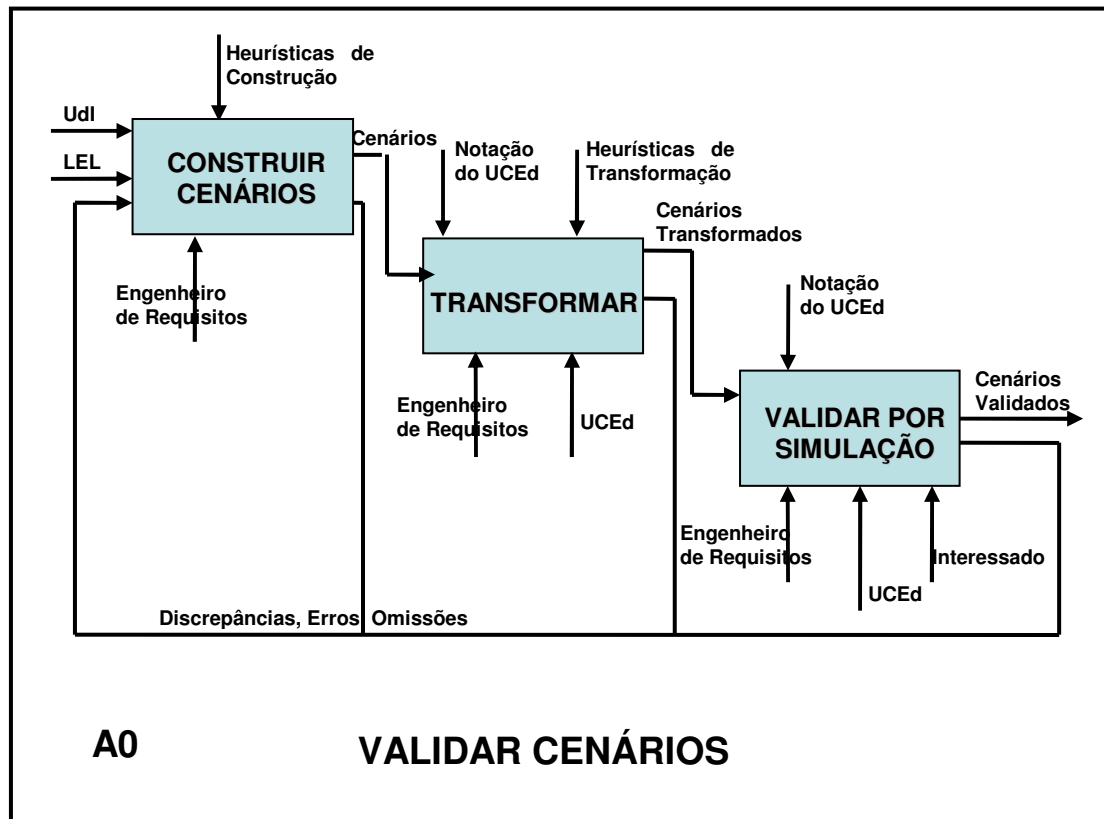


Figura 1: SADT da Estratégia de Validação de Cenários.

O modelo mencionado apresenta as seguintes atividades:

1. CONSTRUIR CENÁRIOS
2. TRANSFORMAR
3. VALIDAR POR SIMULAÇÃO

A estratégia se inicia com a construção dos cenários. Essa construção de cenários segue o processo apresentado por Leite et al. em [1]. Apesar de existir outros processos de construção, optou-se pelo primeiro pelo fato de sua abordagem (*middle-out*) ser diferente de outras abordagens existentes (*top-down* e *bottom-up*). Tal abordagem é resultado de quatro anos de experiências construindo e analisando cenários [1].

Uma vez construídos os cenários, torna-se necessário transformá-los para que estes possam servir como entrada para a ferramenta de simulação. Este trabalho usa especificamente a ferramenta *UCEd* [7],[8], [18].

Em seguida, os cenários serão simulados na ferramenta para que o interessado possa validá-los junto ao engenheiro de requisitos. Nesta parte será colhido o *feedback* fornecido pelo interessado e serão ajustadas possíveis discrepâncias, erros e/ou omissões nos cenários construídos.

4 Detalhamento da Estratégia

Nas seções a seguir, são detalhadas as atividades presentes na estratégia representada no modelo SADT da figura 1.

4.1 CONSTRUIR CENÁRIOS

Como mencionado anteriormente, cenários descrevem situações que ocorrem no Universo de Informações (UdI)². O modelo de cenário adotado neste trabalho é uma estrutura composta pelas entidades: *título, objetivo, contexto, recursos, atores, episódios, exceções e restrições*. A figura a seguir exibe um modelo para descrição de cenários baseada em linguagem natural [1][6]:

<p>Scenario: description of a situation in the application domain. Syntax: Title + Goal + Context + {Resources}₁^N + {Actors}₁^N + {Episodes}₂^N + {Exceptions}</p> <p>Title: identification of the scenario. In the case of a sub-scenario, the title is the same as the episode sentence (see below in the Episode definition), without the constraints. Syntax: Phrase ([Actor Resource] + Verb + Predicate)</p> <p>Goal: aim to be reached in the application domain. The scenario describes the achievement of the goal. Syntax: [Actor Resource] + Verb + Predicate</p> <p>Context: composed by at least one of the following sub-components: Geographical Location: physical set of the scenario. Temporal Location: time specification for the scenario development. Precondition: initial state of the scenario. Syntax: {Geographical Location} + {Temporal Location} + {Precondition} where Geographical Location is: Phrase + {Constraint} where Temporal Location is: Phrase + {Constraint} where Precondition is: [Subject Actor Resource] + Verb + Predicate + {Constraint}</p> <p>Resources: relevant physical elements or information that must be available in the scenario. Syntax: Name + {Constraint}</p> <p>Actors: persons, devices or organization structures that have a role in the scenario. Syntax: Name</p> <p>Episodes: set of actions that details the scenario and provides its behavior. An episode can also be described as a scenario. Syntax (using partial BNF): <episodes> ::= <group series> <episode series> <group series> ::= <group> <group> <non-sequential group> <group series> <group> <group> ::= <sequential group> <non-sequential group> <sequential group> ::= <basic sentence> <sequential group> <basic sentence> <non-sequential group> ::= # <episode series> # <episode series> ::= <basic sentence> <basic sentence> <episode series> <basic sentence> <basic sentence> ::= <simple sentence> <conditional sentence> <optional sentence> <simple sentence> ::= <episode sentence> CR <conditional sentence> ::= IF <condition> THEN <episode sentence> CR <optional sentence> ::= [<episode sentence>] CR where <episode sentence> is described: (([Actor Resource] + Verb + Predicate) ([Actor Resource] + [Verb] + Title)) + {Constraint}</p> <p>Exceptions: usually reflect the lack or malfunction of a necessary resource. An exception hinders the achievement of the scenario goal. The treatment of the exception may be expressed through another scenario. Syntax: Cause [(Solution)] where Cause is: Phrase ([Subject Actor Resource] + Verb + Predicate) where Solution is: Title</p> <p><i>Constraint:</i> a scope or quality requirement referring to a given entity. It is an attribute of Resources, basic Episodes or sub-components of Context. Syntax: ([Subject Actor Resource] + Must [Not] + Verb + Predicate) Phrase</p>
--

+ significa composição, {x} significa zero ou mais ocorrências de x, () é usado para agrupamento, | é usado como OU e [x] denota que x é opcional.

Figura 2: Modelo de Cenário. Retirado de [1].

² O Universo de Informações (ou Universo de Discurso - UofD) inclui todas as fontes de informação e todas as pessoas relacionadas ao software.

Um cenário deve satisfazer um objetivo que é alcançado através de seus episódios. Episódios representam o curso principal da ação, podendo também incluir variações ou alternativas. Durante a execução dos episódios pode ocorrer uma exceção, sinalizando um obstáculo para o alcance do objetivo. A entidade restrição é usada para caracterizar requisitos não funcionais (RNFs)³ aplicados ao contexto, recursos e episódios.

A idéia geral do processo de construção de cenários, detalhada em [1], baseia a descrição dos cenários no vocabulário do Universo de Informação. O processo de construção se inicia com o léxico do domínio da aplicação. Os cenários são posteriormente aperfeiçoados usando-se outras fontes de informação e organizados de maneira a se obter um conjunto consistente de cenários. A figura a seguir, retirada de [2], exhibe o modelo SADT da estratégia de construção de cenários, proposta por [1].

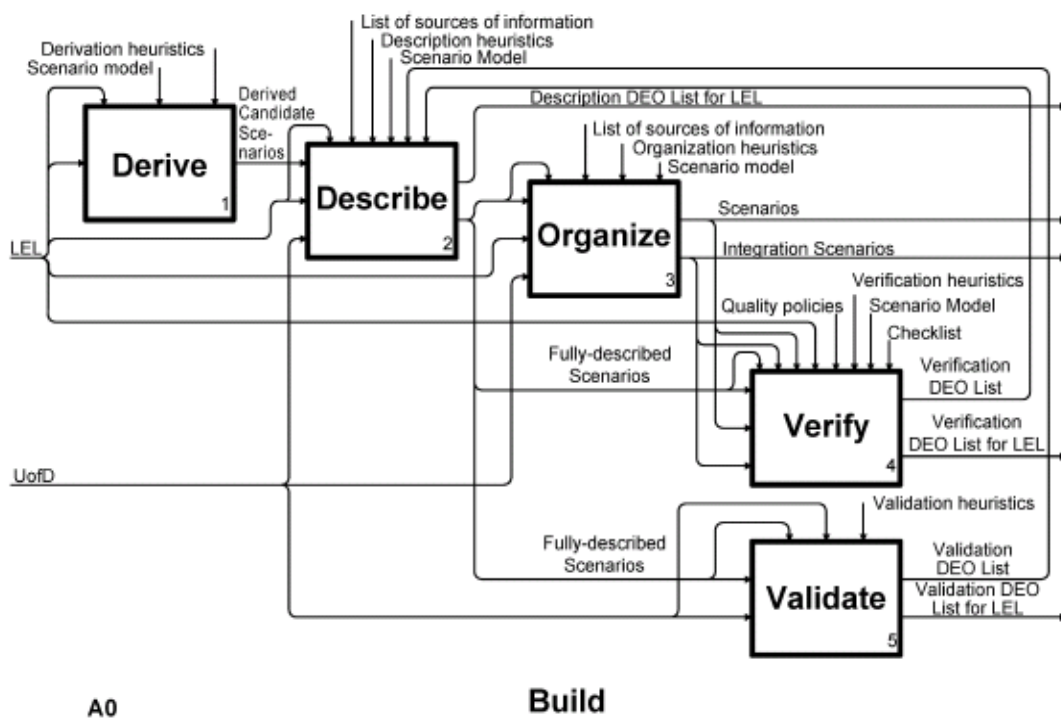


Figura 3: SADT do Processo de Construção de Cenários. Retirado de [2].

As atividades presentes no SADT são descritas abaixo:

1. A atividade *Derive* irá produzir cenários candidatos a partir do léxico, aplicando heurísticas de produção. Tem por objetivo identificar os atores do UdI, identificar cenários candidatos e criar os cenários baseado-se no LEL.
2. A atividade *Describe* refina e complementa os cenários produzidos na etapa anterior, adicionado aos mesmos novas informações provenientes do Universo de

³ Este trabalho refere-se a requisitos não funcionais como metas flexíveis (*Softgoals*) [11-14], [16].

Informações, do processo prévio de elicitação do LEL e do próprio léxico [1]. Durante esta atividade, pode ocorrer a descoberta de novos cenários.

3. A atividade *Organize* é responsável por reorganizar cenários, definindo relações entre eles e dispondos em grupos [1]. Durante a reorganização, ocorrem composições e decomposições de maneira a melhorar a compreensão e o gerenciamento dos cenários [2]. Um cenário é dividido quando ele contém mais de uma situação. Dois ou mais cenários são fundidos quando uma única situação se apresenta artificialmente separada. Em [1] também são identificadas relações na integração de cenários: *hierarchical, overlap, order* e *exception*.
4. A atividade *Verify* é responsável por ajudar na detecção de defeitos (DEOs) dos cenários, tornando-os mais consistentes [1],[2]. Isso é feito através de dois diferentes processos de verificação (*intra-verify* e *inter-verify*) e uma atividade de avaliação (*evaluate*). A atividade *intra-verify* permite uma revisão sistematizada dos cenários individualmente, enquanto a atividade *inter-verify* leva em consideração o conjunto de cenários como um todo. Na atividade *evaluate* têm-se a indicação da aceitação dos cenários ou uma nova verificação.
5. A atividade de *Validate* ajuda a confirmar as informações elicítadas e a detectar defeitos, além disso, pode ocorrer a elicitação de novas informações.

Neste trabalho, a validação será feita junto aos interessados através de simulação, onde os mesmos terão a oportunidade de interagir e discutir as informações elicítadas.

4.2 TRANSFORMAR

Com os cenários construídos em mãos, o engenheiro de requisitos deve transformá-los para que eles possam servir de entrada para a ferramenta *UCEd*.

O *Use Case Editor (UCEd)* [7], [8], [18] é uma ferramenta que permite a construção e edição de casos de uso, a geração de seus respectivos diagramas de estados e a simulação dos mesmos, provendo um suporte automatizado para o engenheiro de requisitos. O *UCEd* foi escolhido para este trabalho por possuir características importantes e únicas comparado a maioria das ferramentas de modelagem de requisitos, além de ser um software livre. Os cenários detalhados na ferramenta podem ser vistos no Apêndice 3.

Porém, como a ferramenta não possui todos os campos presentes no modelo de cenários em [1] e [2], tornou-se necessário elaborar algumas heurísticas de transformação para que nenhum conteúdo do cenário fosse perdido durante a simulação. A seguir, apresentam-se as heurísticas por cada elemento do cenário:

- **Título:** presente no modelo e na ferramenta. Mapeamento feito de forma direta.
- **Objetivo:** presente no modelo e na ferramenta. Mapeamento feito de forma direta.
- **Contexto:** o contexto é composto por: Localização Geográfica, Localização Temporal e Precondição. Tanto Localização Geográfica quanto Localização Temporal não estão presentes explicitamente no programa. Deve-se criar no *Domain Model* um *Concept* Localização Geográfica e um *Concept* Localização

temporal, e adicionar em seus *Possible Values* o valor desejado. Por exemplo, um *Possible Value* para Localização Geográfica pode ser Restaurante [11]. Como sugestão, o engenheiro de requisitos pode explicitar as Localizações Geográficas e Temporais no campo *Description*, para que os mesmos fiquem textualmente destacados no programa.

- **Precondição:** está presente no modelo e na ferramenta, sendo seu mapeamento feito de forma direta. Além disso, deve-se inserir também a Localização Geográfica e Temporal, após declará-las como *Concepts*, de forma a mantê-las inseridas no programa. Um exemplo seria a precondição do cenário “Atendimento da Mesa” [11]: *Mesa is aberta AND Localização Geográfica is Restaurante AND Localização Temporal is durante o atendimento da mesa*.
- **Recursos:** não estão presentes explicitamente na ferramenta. Quando o símbolo de um recurso aparecer nos episódios, possuir estado e/ou restrições, o mesmo deve ser tratado como um *Concept*, sendo declarado no *Domain Model*, bem como devem estar presentes seus estados e/ou restrições. Por exemplo, o recurso *Mesa* foi declarado como um *Concept* e seus estados, *aberta* e *fechada*, como sendo seus *Possible Values*.
- **Atores:** a ferramenta possui para a declaração dos atores, os campos **Primary Actor** e **Participants**. No primeiro campo, insere-se o ator responsável por iniciar o cenário e que deseja ter seu objetivo alcançado. Os demais atores são inseridos no segundo campo.
- **Episódios:** a ferramenta possui o campo **Steps**, onde os episódios devem ser inseridos. É importante ressaltar que a ferramenta suporta todos os tipos de sentenças (sequencial, não sequencial, condicional e optativa).
- **Exceções:** este campo não está explicitamente presente no programa. Neste caso, as exceções são tratadas como passos alternativos (**Alternatives**), dentro do programa. O engenheiro de requisitos fica responsável por escolher na ferramenta em qual episódio será inserido o(s) passo(s) alternativo(s).
- **Restrições:** este campo não está presente explicitamente no programa. Neste caso, serão dados tratamentos diferentes a dois grupos distintos de restrições: restrições comuns e restrições do tipo *softgoals* (metas flexíveis) [11],[12],[13]. As restrições comuns são tratadas como passos alternativos, cabendo ao engenheiro de requisitos inseri-las no episódio mais adequado. As restrições do tipo meta flexíveis são mais complexas, necessitando de um tratamento adequado para sua inserção nos cenários do programa. As restrições do tipo metas flexíveis serão tratadas detalhadamente na próxima seção.

Cabe ressaltar que existem campos na interface do programa que não pertencem aos elementos de um cenário: **System Under Design**, **Primary Actor**, **Participants**, **Invariant**, **Success Post Condition** e **Follow Use Cases**. Os campos **Primary Actor** e **Participants** já foram tratados anteriormente. **System Under Design** pode ser preenchido sem problemas, caso seja conveniente. **Invariant** é uma condição que deve ser mantida através do cenário. Seu preenchimento fica a cargo do engenheiro de requisitos, caso seja necessário, assim como **Success Post Condition**. O campo de extrema importância a ser preenchido é o **Follow Use Cases**, utilizado na ferramenta para ligar os cenários na sequência em que eles ocorrem.

4.2.1 Tratando Restrições do Tipo Metas Flexíveis

Visando a inserção de restrições do tipo metas flexíveis, este trabalho utilizou como base a tese de Cysneiros [14].

O surgimento de metas flexíveis ocorre de maneira natural, durante o levantamento do léxico estendido da linguagem (LEL). Muitas delas ficam em grande destaque em determinados domínios, sendo captados de maneira simples.

Em seguida, devem-se varrer todas as entradas definidas no LEL e comparar cada entrada com alguma base de conhecimento de metas flexíveis, por exemplo [15] e [16]. Ao identificar uma meta flexível relacionada com o símbolo, deve-se inserir essa meta flexível na noção do símbolo, e avaliar os impactos da mesma no símbolo, ou seja, as formas de garantir essa meta flexível, o que poderá gerar novas noções e/ou impactos neste símbolo.

Uma vez que se tem as metas flexíveis expressas no LEL, deve-se agora representá-las de uma forma mais organizada e propícia para lidar com elas. Para isso, será utilizado o grafo de NFRs, proposto por Chung [17], seguindo a seguinte sistematização:

1. Percorrer cada símbolo de LEL verificando quais possuem metas flexíveis.
2. Definir a raiz do RNF como sendo a meta flexível e o tópico como sendo o símbolo do LEL.
3. Decompor a meta em submetas, por tipo ou por tópicos [16].
4. Continuar as decomposições até que se possa visualizar o que é necessário para a operacionalização da meta-flexível.

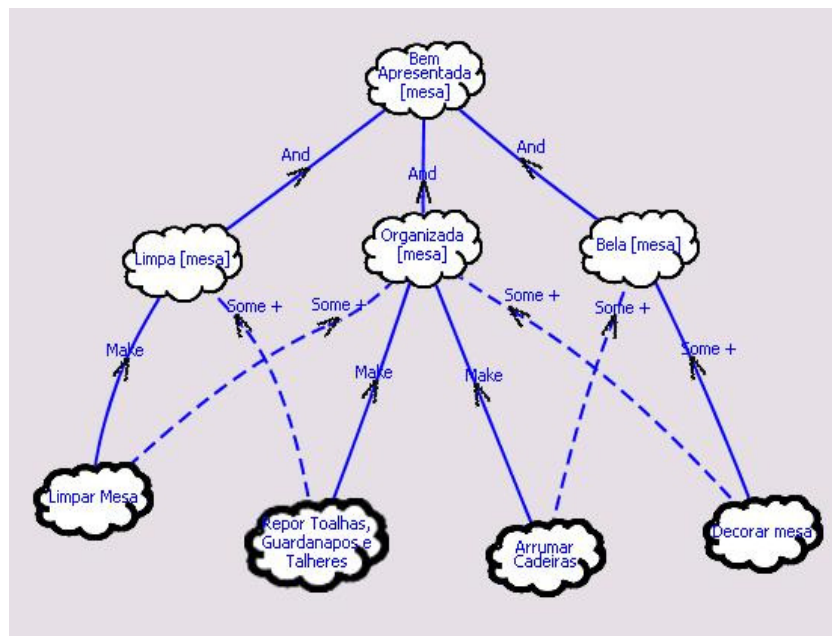


Figura 4: Exemplo de Grafo NFR.

Com as metas flexíveis representadas em seus grafos NFRs, o engenheiro de requisitos deve voltar a sua atenção para a solução de conflitos. Ao se visualizar os grafos de maneira relativamente próxima, pode-se verificar interdependências positivas ou negativas e assim resolver problemas de conflitos que venham a ocorrer.

Para que o engenheiro de requisitos não dependa que a organização dos grafos favoreça a identificação de interdependências, [14] propõe as seguintes heurísticas:

1. Comparar todos os grafos de um mesmo tipo. Exemplo: comparar todos os grafos relativos a meta flexível Performance.
2. Comparar grafos de tipos possivelmente conflitantes. Exemplo: grafos que envolvam Segurança e Desempenho.
3. Comparar os grafos por pares, de forma a comparar um dado grafo com todos os outros, sem repetir as comparações feitas anteriormente.

A figura seguinte [14] mostra um exemplo de metas flexíveis conflitantes. O engenheiro de requisitos deve procurar determinar que desenhos alternativos são possíveis para contornar o conflito ou se uma meta flexível será satisfeita em detrimento das outras.

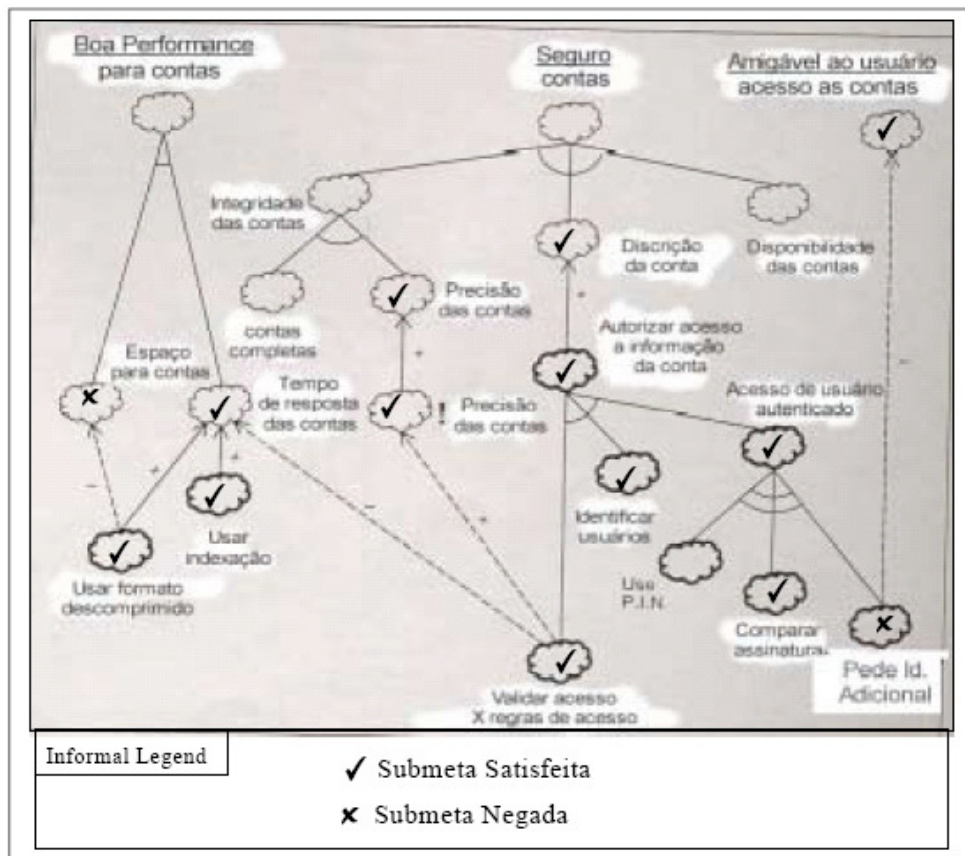


Figura 5: Exemplo de Metas Flexíveis Conflitantes. Retirada de [14].

O próximo passo é integrar as metas flexíveis operacionalizadas aos cenários. Uma vez escolhido o cenário, deve-se verificar que símbolos do LEL aparecem no título deste cenário. Para cada símbolo, deve-se verificar se no conjunto de grafos NFRs existe algum onde o símbolo aparece. Se existir, o engenheiro de requisitos deve avaliar cada uma das operacionalizações para ver se elas se aplicam ao cenário analisado. Caso se

apliquem, deve-se verificar o cenário para se certificar da existência de episódios que garantam a satisfação dessas operacionalizações.

Se os episódios existentes não forem suficientes para garantir a satisfação da meta flexível, o engenheiro de requisitos deve incluir 1 (um) ou mais episódios de forma a satisfazer esta meta flexível. Ao término das inclusões necessárias, deve-se analisar a ocorrência de alguns conflitos que possam vir a ocorrer. Caso um conflito venha a ocorrer, o engenheiro de requisitos deve retornar ao grafo NFR para verificar o quanto prejudicial seria a não satisfação desta meta flexível, e avaliar junto ao interessado possibilidades para contornar o conflito.

Ao inserir novos episódios nos cenários do programa *UCEd*, é imperativo que o engenheiro de requisitos coloque após o episódio inserido a seguinte expressão: //Nome_da_Meta_Flexível [Símbolo]. Essa informação é importante para que a rastreabilidade no modelo seja mantida. As duas barras no início da expressão são para que o programa considere a mesma como um comentário dentro do cenário, evitando a influência do texto na compilação deste cenário.

A seguir serão exibidos exemplos de um cenário com metas flexíveis em separado e posteriormente, com suas respectivas operacionalizações inseridas nos episódios na ferramenta.

Título:	ATENDIMENTO DA MESA
Objetivo:	Restaurante possa faturar
Contexto:	
Localização geográfica:	Restaurante
Localização temporal:	Durante o atendimento da mesa.
Precondição:	Mesa está aberta.
Recursos:	comanda em branco, mesa, menu opções e escaninho
Atores:	Cliente e garçom
Episódios:	Cliente faz pedido. Cliente pergunta sobre opção. Garçom responde sobre opção. Cliente escolhe opção. Garçom atende a mesa. Garçom anota pedido. Garçom joga a comanda.
Restrição:	Para transferir de mesa para mesa, a mesa de destino precisa estar fechada.
Exceções:	Se cliente desejar trocar de mesa: (Caixa deve transferir de mesa para mesa)
Metas flexíveis:	Satisfeito [cliente], Corretas[comandas], Aumento[atendimento]

Figura 6: Cenário Original com Metas Flexíveis Identificadas em Separado.

Retirado de [11].

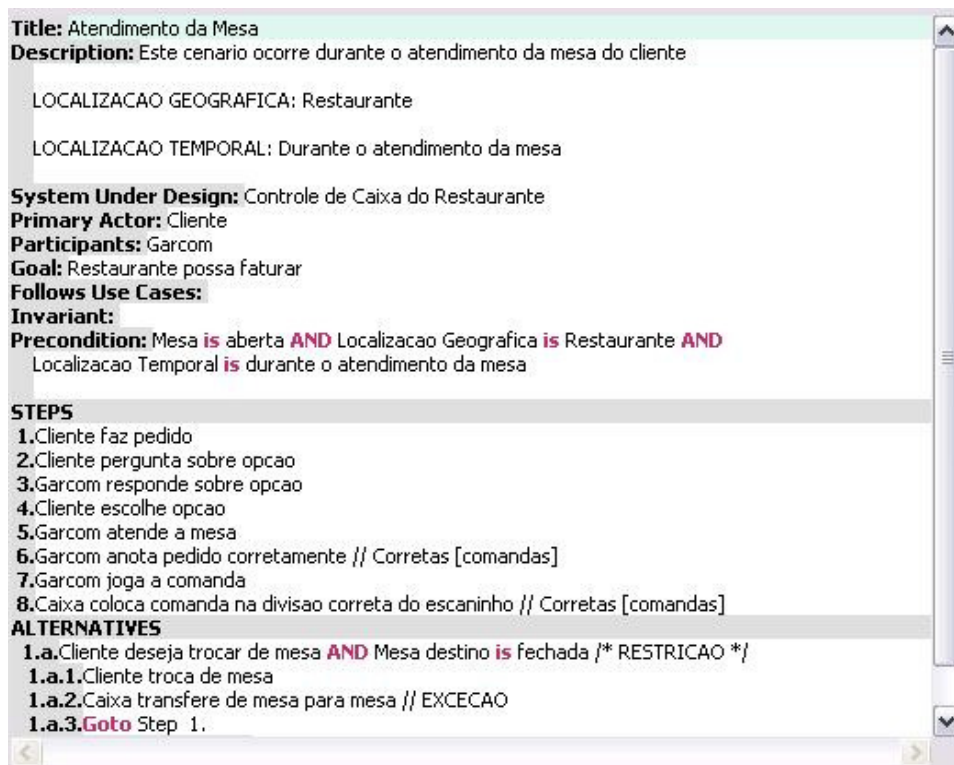


Figura 7: Operacionalizações Inseridas nos Cenários na Ferramenta UCEd.

É válido observar alguns detalhes importantes na figura 5. Como mencionado anteriormente, após inserir novos episódios, é necessário inserir o comentário com a meta flexível que originou o novo episódio, mantendo assim a rastreabilidade do modelo. Esse recurso também é de grande valia para manter a rastreabilidade para exceções e restrições comuns.

Ao término desses passos, o cenário está totalmente adaptado à ferramenta, podendo assim ser simulado e conseqüentemente validado pelos interessados.

4.3 VALIDAR POR SIMULAÇÃO

Nesta etapa ocorre a formalização da validação dos requisitos elicitados. O engenheiro de requisitos deverá preparar a ferramenta para a simulação dos episódios com o interessado.

O ambiente que ocorrerá essa simulação é um ambiente controlado, com registro em áudio dos comentários do próprio interessado e gravação também da tela em que a validação estiver ocorrendo, tudo previamente autorizado por um Termo de Consentimento. Além disso, o engenheiro de requisitos deve estar ao lado do interessado, para coletar dados adicionais e auxiliá-lo na utilização da ferramenta. A figura a seguir mostra um exemplo de janela de simulação do programa *UCEd*.

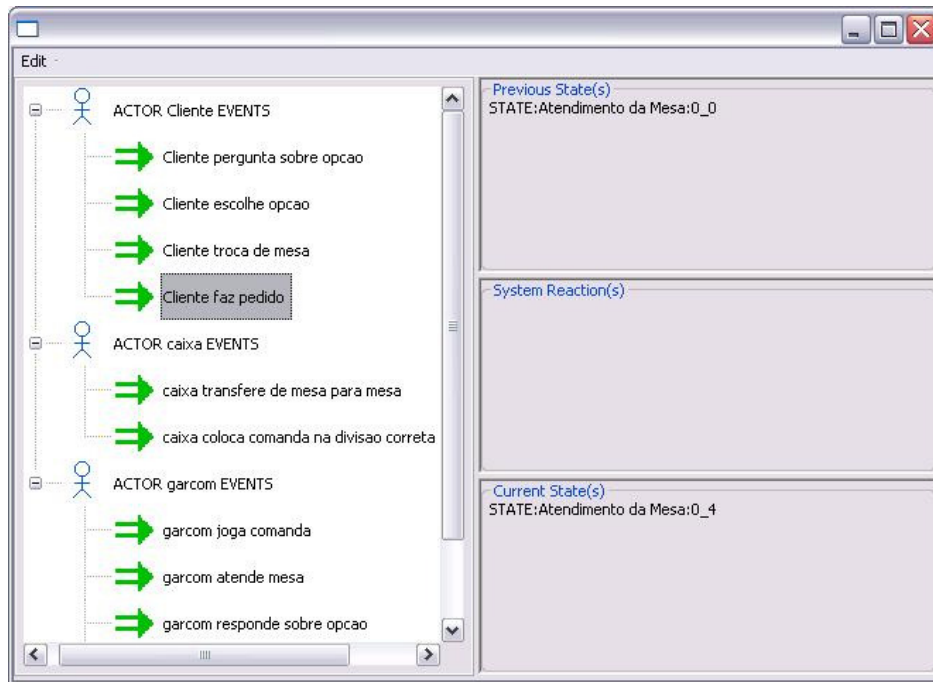


Figura 7: Simulação do Cenário Atendimento da Mesa.

A validação realizada estará concentrada nos seguintes tópicos:

1. Léxico
2. Ordem dos Episódios
3. Completude e Corretude dos Episódios

No caso do léxico, é importante o engenheiro de requisitos observar o surgimento de algum símbolo novo (podendo ser sinônimo ou não), ou se algum símbolo existente teve seu significado alterado.

A ordem inicial dos episódios pode ser alterada pelo interessado. Esse fato pode ser observado pelo engenheiro de requisitos durante a seleção de episódios na ferramenta. Caso seja selecionado um episódio fora da sequência inicial, a ferramenta exibe uma janela chamando a atenção do fato (figura 9), e o engenheiro de requisitos deve se certificar junto ao interessado se ele deseja propositalmente trocar a ordem do episódio selecionado ou não. Cabe considerar que a mera troca de ordem dos episódios não causa impacto direto no léxico.

A completude e corretude dos episódios serão observadas pelo engenheiro de requisitos durante a simulação ou após uma minuciosa análise dos registros de vídeo (do interessado e da tela de simulação), de forma a evitar ao máximo a perda de informações. Ao analisar completude e corretude dos episódios, o engenheiro de requisitos deve ficar atento a 3 (três) casos que podem ocorrer: a criação de novos episódios, a eliminação de episódios e a união de episódios. Em todos esses casos podem ocorrer impactos no léxico do domínio da aplicação.

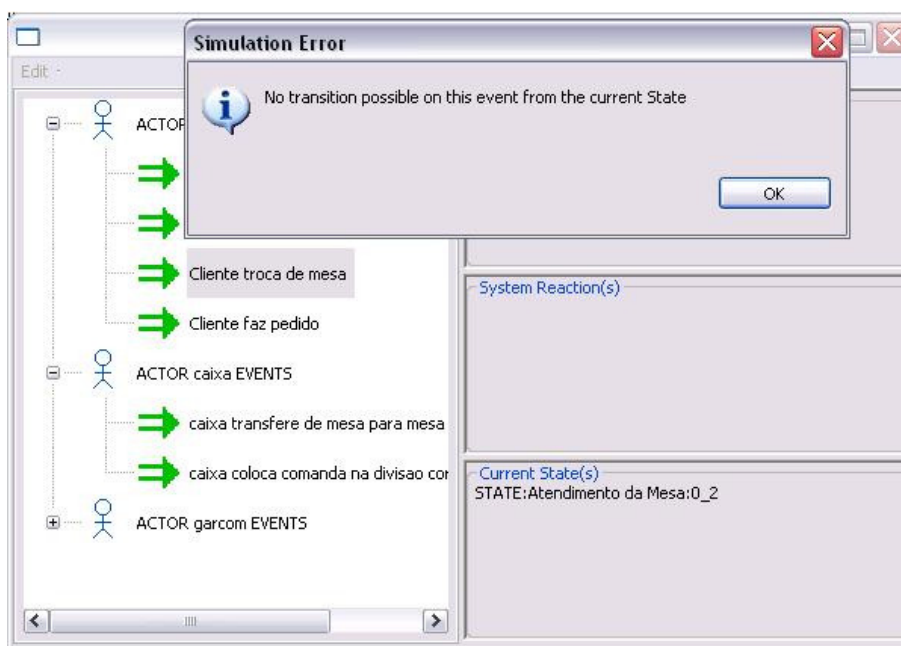


Figura 8: Escolha de um Episódio Diferente da Ordem Inicial.

Ao término da simulação, o engenheiro de requisitos deverá analisar todo o material procurando discrepâncias, erros e/ou omissões, conforme o foco mostrado anteriormente. Erros e omissões são detectados principalmente durante a simulação com o interessado. Discrepâncias são encontradas principalmente na análise posterior do material. Pode ocorrer também o surgimento de novos cenários.

As alterações encontradas no modelo inicial devem ser feitas pelo engenheiro de requisitos nos cenários construídos e uma nova simulação deve ser realizada, visando o refinamento dos modelos.

No final desta etapa, pode-se confirmar se as situações pertencentes ao Universo de Informação foram relatadas adequadamente nos cenários construídos.

5 Um Estudo Preliminar da Estratégia

Para validar a estratégia proposta foi realizado um estudo preliminar no laboratório do *Semiotic Engineering Research Group* – SERG, na PUC-Rio. Para tal, foi escolhido um projeto de um sistema de Controle de Caixa de Restaurante [11] e quatro voluntários com conhecimento do domínio fazendo o papel de interessados.

O sistema de Controle de Caixa de Restaurante foi descrito utilizando 4 cenários (Apêndice 2): Liberação da Mesa, Atendimento da Mesa, Fechamento da Conta e Rateio dos 10%.

O objetivo do estudo era comprovar a eficácia e eficiência da estratégia. Sendo assim, dividiu-se o estudo em duas partes: uma parte escrita, utilizando um Formulário de Validação (Apêndice 1), onde os episódios estão previamente ordenados e a outra parte utilizando a ferramenta de simulação onde os episódios estão fora de ordem. A simulação foi gravada utilizando um *software* de captura de tela e áudio.

Os interessados foram divididos em dois grupos G1 e G2, cada grupo com 2 componentes, para realizar a validação de todos os cenários. O grupo G1 realizou primeiro a parte escrita com formulários (Apêndice 1) e por último a parte da ferramenta de simulação. O grupo G2 seguiu a ordem inversa. Cabe ressaltar que as partes do teste eram mediadas pelo engenheiro de requisitos, onde o mesmo procurava sempre não influenciar a opinião dos interessados.

A análise dos resultados levou em consideração as alterações realizadas pelos grupos nos cenários validados. Considera-se alteração qualquer uma das seguintes alternativas:

- Criação de um novo episódio
- Correção de um episódio existente
- Ordenação entre episódios (sem envolver mudança no léxico)
- Eliminação de algum episódio
- Fusão entre episódios

As alterações obtidas pelos grupos foram somadas tanto na parte do formulário quanto na parte da ferramenta. Os resultados do grupo G1 são descritos pelos gráficos a seguir:

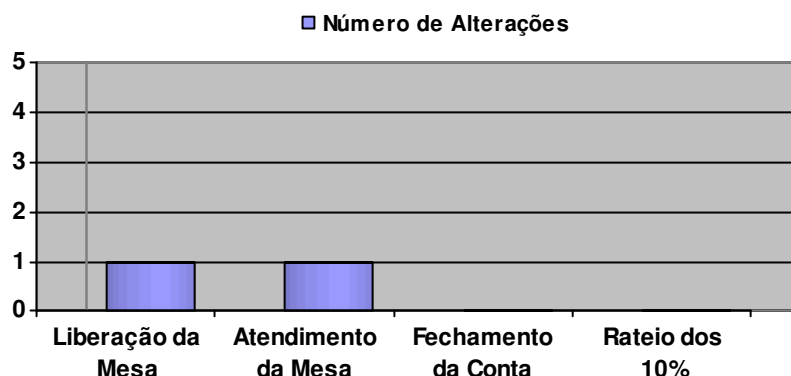


Figura 9: Total de Alterações Realizadas pelo Grupo G1 utilizando Formulário.

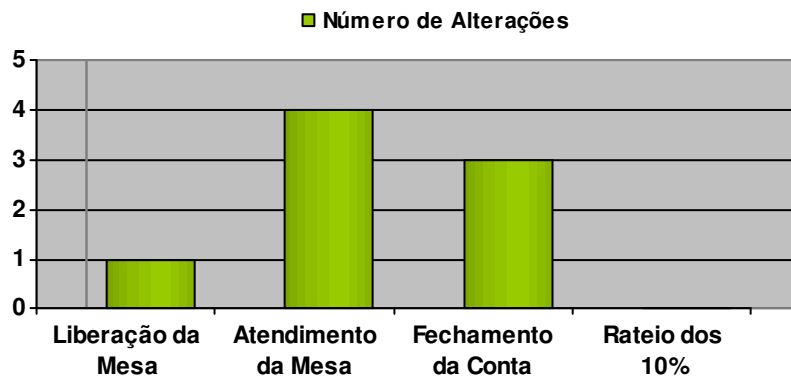


Figura 10: Total de Alterações Realizadas pelo Grupo G1 utilizando a Ferramenta.

O tipo de comportamento apresentado pelos gráficos da figura 10 e 11 pode ser explicado utilizando o conceito de validação ativa e passiva. Nos formulários de validação, os episódios estão previamente ordenados, utilizando o raciocínio e o conhecimento do engenheiro de requisitos. Neste caso, o interessado assume uma postura passiva na maioria das vezes, dizendo se concorda ou não com o que foi apresentado e fazendo alterações nos pontos que chamaram muito a sua atenção.

Já na utilização da ferramenta de simulação, por não haver uma ordenação inicial definida, o interessado é de certa forma obrigado a propor uma ordenação que será validada ou não pela simulação, colocando-se algumas vezes em problemas e situações que não foram percebidas durante a validação com episódios pré-ordenados. Neste caso, o interessado age ativamente na validação dos episódios, sendo de certa forma o responsável pela elaboração dos cenários. Além disso, através das gravações, o engenheiro de requisitos consegue captar mínimos detalhes da validação, permitindo uma validação mais rica e detalhada.

Os resultados do grupo G2 são apresentados através dos seguintes gráficos:

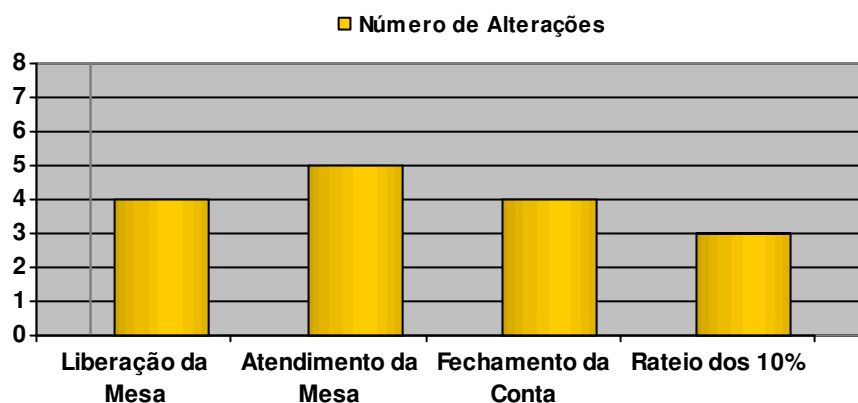


Figura 11: Total de Alterações Realizadas pelo Grupo G2 utilizando Formulário.

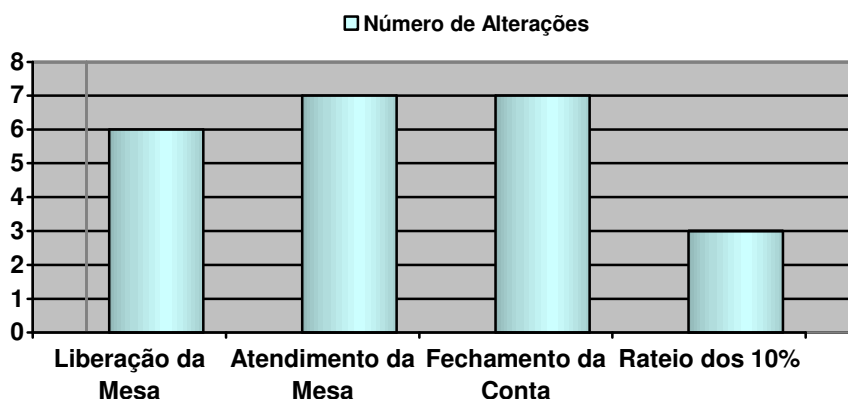


Figura 12: Total de Alterações Realizadas pelo Grupo G2 utilizando a Ferramenta.

É fato observar que o grupo G2 encontrou um maior número de alterações em relação ao grupo G1, tanto na parte de formulário quanto na parte da ferramenta. Isso pode ser explicado baseando-se na ordem de execução do estudo. O grupo G1 teve contato inicialmente com o formulário de validação, enquanto o primeiro contato do grupo G2 foi diretamente com a ferramenta.

Com isso, os integrantes do grupo G1 observaram o pré-ordenamento dos episódios antes de utilizar a ferramenta, o que pode ter colaborado para a obtenção de um menor número de alterações durante o uso da mesma. Já os integrantes do G2 não tiveram qualquer contato prévio com o pré-ordenamento, melhorando assim o raciocínio e a capacidade de questionamento dos interessados do grupo durante a validação. Isso se refletiu nos formulários de validação preenchidos pelo grupo C2 logo após a sessão de validação com a ferramenta de simulação. Tais formulários carregaram em sua maioria alterações idênticas as identificadas durante o uso anterior da ferramenta de simulação.

Ao término do estudo, os interessados de ambos os grupos foram perguntados a respeito da utilização da estratégia de validação utilizando uma ferramenta de simulação. A maioria dos integrantes confirmou que a utilização de uma ferramenta onde o interessado age de forma ativa, como na ferramenta de simulação apresentada, contribui de maneira mais eficaz na validação das informações elicitadas. Foram obtidas respostas do tipo: “a utilização da estratégia permite ao validador se colocar dentro do cenário” e também “é um novo jeito de pensar e agir, mantendo o validador raciocinando e analisando o tempo todo”. Também foram obtidas sugestões para possíveis melhorias na ferramenta.

6 Conclusões

Este trabalho apresentou o desenvolvimento de uma estratégia para validação de cenários baseado em técnicas interativas, como simulação e registro de áudio e de imagens.

A estratégia proposta foi descrita em três atividades: Construir Cenários, visando uma melhor elaboração dos mesmos; Transformar, atividade necessária para adequar os cenários construídos a ferramenta de simulação; e Validar por Simulação, que é a formalização da atividade de validação com o uso da ferramenta de simulação e outras técnicas interativas.

Em seguida foi realizado um estudo preliminar com dois grupos de interessados, comparando o desempenho da validação utilizando formulários e a estratégia de simulação. Tal estudo teve resultados bastante motivadores, onde a estratégia proposta conseguiu se mostrar uma boa opção para a validação de cenários.

É importante afirmar que a estratégia de validação baseada em simulação se torna ainda mais eficiente quanto melhor forem realizadas a elicitação e modelagem de requisitos, servindo como ajuste fino no entendimento do engenheiro de requisitos do domínio do problema.

Como trabalhos futuros, são sugeridos o refinamento da estratégia, buscando melhores heurísticas para o diálogo com o interessado no momento da validação; e a melhoria da ferramenta de simulação, tornando-a mais adaptada a trabalhar com cenários.

Referências

- [1] LEITE, J. C. S. P.; HADAD, G. D. S.; DOORN, J. H.; KAPLAN, G. N. A Scenario Construction Process. *Requirements Engineering Journal* 5(1): 38-61, 2000.
- [2] LEITE, J. C. S. P.; DOORN, J. H.; HADAD, G. D. S.; KAPLAN, G. N. Scenario Inspections. *Requirements Engineering Journal* 10: 1-21, 2005.
- [3] FAGAN M. E. Design and Code Inspections to Reduce Erros in Program Development. *IBM Systems Journal*, 15(3): 182-211, 1976.
- [4] BREITMAN, K. K.; LEITE, J. C. S. P. A Framework for Scenario Evolution. Proc. of the THIRD IEEE INTERNATIONAL CONFERENCE ON REQUIREMENTS ENGINEERING. IEEE Computer Society Press, Los Alamitos, CA, 1998, pp:214-221.
- [5] ROSS, D.; SCHOMAN, A. Structured Analysis for Requirements Definition. In: IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, 3(1): 6-15, 1977.
- [6] LEITE, J.C.S.P.; ROSSI, G.; BALAGUER, F.; MAIORANA, V.; KAPLAN, G.; HADAD, G.; OLIVEROS, A. Enhancing a Requirements Baseline with Scenarios. *Requirements Engineering Journal*, 1997; 2(4): 184-198.
- [7] SOMÉ, S. Supporting Use Cases Based Requirements Simulation. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING RESEARCH AND PRACTICE (SERP'04), June 2004.
- [8] SOMÉ, S. An Environment for Use Cases based Requirements Engineering. In: 12th IEEE INTERNATIONAL CONFERENCE ON REQUIREMENTS ENGINEERING (RE 2004), September 2004.
- [9] FLEISCH, W. Applying Use Cases for the Requirements Validation of Component-Based Real Time Software. In Proceedings of 2nd IEEE INTERNATIONAL SYMPOSIUM ON OBJECT-ORIENTED REAL-TIME DISTRIBUTED COMPUTING (ISORC). Saint-Malo, France: IEEE Computer Society, May 1999, pp. 75-84.
- [10] PREECE, J.; YVONNE, R.; SHARP, H. **Design de Interação**. Bookman Editora, ISBN 85-363-0494-4, 2005.
- [11] OLIVEIRA, A. P. A. **Engenharia de Requisitos Intencional: Um Método de Elicitação, Modelagem e Análise de Requisitos**. 2008. Tese de Doutorado. Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2008.
- [12] YU, E. **Modeling Strategic Relationships for Process Reengineering**. Ph.D. Thesis, Graduate Dept. of Computer Science, University of Toronto, 1995.
- [13] CUNHA, H. S. **Uso de Estratégias Orientadas a Metas para Modelagem de Requisitos de Segurança**. Dissertação de Mestrado, Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, Março de 2000.
- [14] CYSNEIROS, L. M. **Requisitos Não Funcionais: Da Elicitação ao Modelo Conceitual**. Tese de Doutorado, Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, Fevereiro de 2001.
- [15] NETO, J. S. M. **Integrando Requisitos Não Funcionais ao Modelo de Objetos**. Dissertação de Mestrado, Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, Março de 2000.
- [16] CHUNG, L.; NIXON, B.; YU, E.; MYLOPOULOS, J. **Non-Functional Requirements in Software Engineering**. Kluwer Academic Publishers, 1999.

[17] CHUNG, L. **Representing and Using Non-Functional Requirements: A Process Oriented Approach**. Ph.D. Thesis, Dept. of Computer Science, University of Toronto, June 1993.

[18] Use Case Editor. Disponível em: <http://www.site.uottawa.ca/~ssome/Use_Case_Editor_UCEd.html>. Acesso em: 10 Ago. 2009.

[19] GOUGH, P. A.; FODEMSKI, F.T. Scenarios - an Industrial Case Study and Hypermedia Enhancements. In: Proceedings of the IEEE INTERNATIONAL SYMPOSIUM ON REQUIREMENTS ENGINEERING, IEEE, Computer Society Press, 1995, pp 10-17.

[20] HAUMER, P.; HEYMANS, P. JARKE, M.; POHL, K. Bridging the Gap Between past and Future in RE: A Scenario-Based Approach. In: Proceedings 4th IEEE INTERNATIONAL SYMPOSIUM ON REQUIREMENTS ENGINEERING, IEEE Computer Society Press, 1995, 66-73.

Apêndice 1 Exemplo de Formulário de Validação

FORMULÁRIO DE VALIDAÇÃO DE CENÁRIO		
CENÁRIO 2/4: ATENDIMENTO DA MESA		
DESCRIÇÃO: Este cenário ocorre durante o atendimento da mesa do cliente		
DEFINIÇÕES	NADA A ACRESCENTAR	SUGESTÕES PARA ALTERAÇÕES
LOCALIZAÇÃO GEOGRÁFICA: Restaurante		
LOCALIZAÇÃO TEMPORAL: Durante o atendimento da mesa		
PRECONDIÇÃO: Mesa está aberta		
RECURSOS: Comanda em branco		
Mesa		
Menu		
Opções		
Escaninho		
ATORES: Cliente		
Garçom		
OBJETIVO: Restaurante possa faturar		
EPISÓDIOS	SUGESTÕES PARA ALTERAÇÕES	
1. Cliente faz pedido		
2. Cliente pergunta sobre opção		

3. Garçom responde sobre opção	
4. Cliente escolhe opção	
5. Garçom atende a mesa	
6. Garçom anota pedido corretamente	
7. Garçom joga a comanda	
8. Caixa coloca comanda na divisão correta do escaninho	
CURSOS ALTERNATIVOS	SUGESTÕES PARA ALTERAÇÕES
1.a Cliente deseja trocar de mesa e mesa destino está fechada	
1.a.1. Cliente troca de mesa	
1.a.2. Caixa transfere de mesa para mesa	

Apêndice 2 Cenários Sistema de Controle de Caixa de Restaurante

Título: LIBERAÇÃO DA MESA

Objetivo: Mesas estejam faturando

Contexto:

Localização geográfica: Restaurante

Localização temporal: Na abertura do dia ou depois de fechar a mesa.

Precondição: Mesa está fechada.

Recursos: escaninho, toalha, guardanapos e talheres

Atores: Caixa e garçom

Episódios: Caixa abre a mesa.

Garçom prepara a mesa.

Garçom limpa a mesa.

Garçom repõe toalha, guardanapos e talheres.

Restrição: Somente a mesa fechada pode ser aberta.

Exceções: Não há.

Metas flexíveis: Bem apresentada [mesa]

Título: ATENDIMENTO DA MESA

Objetivo: Restaurante possa faturar

Contexto:

Localização geográfica: Restaurante

Localização temporal: Durante o atendimento da mesa.

Precondição: Mesa está aberta.

Recursos: comanda em branco, mesa, menu opções e escaninho

Atores: Cliente e garçom

Episódios: Cliente faz pedido.

Cliente pergunta sobre opção.

Garçom responde sobre opção.

Cliente escolhe opção.

Garçom atende a mesa.

Garçom anota pedido.

Garçom joga a comanda.

Restrição: Para transferir de mesa para mesa, a mesa de destino precisa estar fechada.

Exceções: Se cliente desejar trocar de mesa:

(Caixa deve transferir de mesa para mesa)

Metas flexíveis: Satisfeito [cliente], Corretas[comandas], Aumento[atendimento]

Título: FECHAMENTO DA CONTA
Objetivo: **Mesa esteja (novamente) aberta**
Contexto:
Localização geográfica: Restaurante
Localização temporal: Após a solicitação do cliente ou no encerramento do funcionamento do restaurante.
Precondição: Mesa está aberta e sendo atendida.
Recursos: comandas, bordêro e calculadora
Restrição: comandas e bordêro devem ter o número da mesa
Atores: Cliente, garçom e caixa
Episódios: Cliente pede a conta.
Caixa calcula a soma das comandas e anota no controle interno.
Caixa anota os 10% no canhoto.
Caixa faz controle do caixa (alocação das comandas).
Caixa entrega conta ao garçom.
*Garçom confere as comandas anexadas.
Cliente paga a conta.
Garçom entrega pagamento ao caixa.
Restrição: *Se garçom encontrar erro ele devolve ao caixa.
Exceções: Não há.
Metas flexíveis: Satisfeito [cliente], Honesto[pagamento], Correta[conta], Sem erro[pagamento]

Título: RATEIO DOS 10 %
Objetivo: **Garçom seja gratificado**
Contexto:
Localização geográfica: Restaurante
Localização temporal: No final do dia.
Precondição: Todas as mesas foram fechadas.
Recursos: canhotos e calculadora
Atores: Caixa e garçom
Episódios: Caixa divide o valor dos canhotos pelos garçons.
Garçom recebe o rateio dos 10%.
Restrição: So acontece se restaurante cobrar 10%.
Exceções: Não há.
Metas flexíveis: Correto [10%]

Apêndice 3 Cenários Transformados para a Ferramenta

Title: Liberacao da Mesa

Description: Este cenario ocorre quando a mesa esta fechada e necessita ser liberada para receber novos clientes

LOCALIZACAO GEOGRAFICA: Restaurante

LOCALIZACAO TEMPORAL: Na abertura do dia ou depois de fechar a mesa

System Under Design: Controle de Caixa do Restaurante

Primary Actor: Caixa

Participants: Garcom

Goal: Mesas estejam faturando

Follows Use Cases:

Invariant:

Precondition: Mesa **is** fechada **AND** Localizacao Geografica **is** Restaurante **AND**

Localizacao Temporal **is** abertura do dia **OR** Localizacao Temporal **is** depois de fechar a mesa

STEPS

- 1.Caixa abre a mesa
- 2.Garcom prepara a mesa
- 3.Garcom limpa a mesa
- 4.Garcom repoe toalha, guardanapos e talheres
- 5.Garcom arruma as cadeiras // Bem Apresentada [mesa]

ALTERNATIVES

- 1.a.If mesa **is not** fechada /* RESTRICAO */
 - 1.a.1.Caixa aguarda o fechamento da mesa
 - 1.a.2.**Goto** Step 1.

Success Postcondition: Mesa **is** aberta

Title: Atendimento da Mesa

Description: Este cenario ocorre durante o atendimento da mesa do cliente

LOCALIZACAO GEOGRAFICA: Restaurante

LOCALIZACAO TEMPORAL: Durante o atendimento da mesa

System Under Design: Controle de Caixa do Restaurante

Primary Actor: Cliente

Participants: Garcom

Goal: Restaurante possa faturar

Follows Use Cases:

Invariant:

Precondition: Mesa **is** aberta **AND** Localizacao Geografica **is** Restaurante **AND**

Localizacao Temporal **is** durante o atendimento da mesa

STEPS

- 1.Cliente faz pedido
- 2.Cliente pergunta sobre opcao
- 3.Garcom responde sobre opcao
- 4.Cliente escolhe opcao
- 5.Garcom atende a mesa
- 6.Garcom anota pedido corretamente // Corretas [comandas]
- 7.Garcom joga a comanda
- 8.Caixa coloca comanda na divisao correta do escaninho // Corretas [comandas]

ALTERNATIVES

- 1.a.Cliente deseja trocar de mesa **AND** Mesa destino **is** fechada /* RESTRICAO */
 - 1.a.1.Cliente troca de mesa
 - 1.a.2.Caixa transfere de mesa para mesa // EXCECAO
 - 1.a.3.**Goto** Step 1.

Success Postcondition: Mesa **is** atendida

Title: Fechamento da Conta**Description:** Este cenario ocorre quando o cliente deseja fechar sua conta

LOCALIZACAO GEOGRAFICA: Restaurante

LOCALIZACAO TEMPORAL: Apos a solicitacao do cliente ou no encerramento do funcionamento do restaurante

System Under Design: Controle de Caixa do Restaurante**Primary Actor:** Cliente**Participants:** Garcom **AND** Caixa**Goal:** Mesa esteja novamente aberta**Follows Use Cases:****Invariant:****Precondition:** Mesa **is** aberta **AND** Mesa **is** sendo atendida **AND** Localizacao Geografica **is** Restaurante **AND** Localizacao Temporal **is** apos solicitacao do cliente **OR** Localizacao Temporal **is** encerramento do funcionamento do restaurante**STEPS**

1. Cliente pede a conta
2. Caixa calcula a soma das comandas e anota **no** controle interno
3. Caixa anota os 10 por cento **no** canhoto
4. Caixa faz controle do caixa alocando as comandas
5. Caixa confere conta // Correta [conta]
6. Caixa entrega conta ao garcom
7. Garcom confere as comandas anexadas // Corretas [comandas]
8. Cliente confere conta // Correta [conta]
9. Cliente confere valor do pagamento com valor da conta // Honesto [pagamento]
10. Cliente paga a conta
11. Garcom entrega pagamento ao caixa
12. Caixa confere valor do pagamento com valor da conta // Sem Erro [pagamento]

ALTERNATIVES

- 7.a. **If** garcom encontra erro nas comandas /* RESTRICAO */
 - 7.a.1. Garcom devolve comandas ao caixa
 - 7.a.2. **Goto** Step 2.

Success Postcondition: conta **is** fechada**Title:** Rateio dos 10 Porcento**Description:** este cenario ocorre quando se deseja gratificar os garcons com o rateio dos 10%

LOCALIZACAO GEOGRAFICA: Restaurante

LOCALIZACAO TEMPORAL: No final do dia

System Under Design: Controle do Caixa do Restaurante**Primary Actor:** Caixa**Participants:** Garcom**Goal:** Garcom seja gratificado**Follows Use Cases:****Invariant:****Precondition:** Mesas **is** fechadas **AND** Localizacao Geografica **is** Restaurante **AND** Localizacao Temporal **is** final do dia**STEPS**

1. Caixa divide o valor dos canhotos pelos garcons
2. Caixa confere calculo dos 10 por cento // Correto [10%]
3. Garcom recebe o rateio dos 10 por cento
4. Garcom confere calculo dos 10 por cento // Correto [10%]

ALTERNATIVES

- 1.a. Restaurante nao cobra 10 por cento /* RESTRICAO */
 - 1.a.1. Garcom nao recebe rateio dos 10 por cento

Success Postcondition: Garcom **is** gratificado

Apêndice 4 Léxico Sistema Controle do Caixa de Restaurante

Projeto:	CONTROLE DO CAIXA DE RESTAURANTE
Data de criação:	19-12-2006
Descrição:	Linguagem do domínio da aplicação para o controle do caixa de restaurantes.

Símbolos do Tipo Sujeito

Nome:	caixa
Noção:	1 - pessoa que trabalha no restaurante fazendo o controle das mesas.
Classificação:	sujeito
Impacto(s):	1 - realiza as tarefas: abrir a mesa , botar a comanda no escaninho , transferir de mesa para mesa , fechar a mesa , retirar comandas do escaninho , calcular o total , conferir pagamento .
Sinônimo(s):	caixas.

Nome:	cliente
Noção:	1 - Pessoa que consome opção do restaurante .
Classificação:	sujeito
Impacto(s):	1 - senta a mesa , faz pedido , troca de mesa , pede conta , paga conta .
Sinônimo(s):	fregues, clientes.

Nome:	garçom
Noção:	1 - Pessoa que trabalha no restaurante servindo o cliente . 2 - Responsável pela comunicação entre os clientes e o caixa .
Classificação:	sujeito
Impacto(s):	1 - Realiza as tarefas: jogar a comanda , entregar o pagamento .
Sinônimo(s):	garçons.

Símbolos do Tipo Objeto

Nome:	canhoto
Noção:	1 - só existe se o restaurante cobra 10%. 2 - papel onde o caixa anota os 10% no canhoto, quando fecha a mesa .
Classificação:	objeto
Impacto(s):	1 - ao final do dia, o garçon recebe o valor dos canhotos. 2 - ao final do dia, o valor dos canhotos é dividido pelos garçons . 3 - o caixa anota os 10% no canhoto, quando fecha a mesa .
Sinônimo:	canhotos.

Nome:	10%
Noção:	1 - Igual a 1/10 do total . 2 - Esta adicionado ao total se o restaurante cobra 10% .
Classificação:	objeto
Impacto(s):	1 - Eh anotado no canhoto , quando o caixa fecha a mesa .
Sinônimo(s):	gorgeta.

Nome:	bordero
Noção:	1 - É uma tabela que contém a data e a unidade. 2 - Contém uma lista de num. do carimbo , num. da mesa e quantidade de comandas.
Classificação:	objeto
Impacto(s):	1 - Quando o caixa fecha a mesa anota o num. da mesa e a quantidade de comandas no bordero.
Sinônimo(s):	

Nome:	comanda
Noção:	1 - O garçom anota os pedidos e o num. da mesa do cliente na comanda, quando o cliente faz o pedido . 2 - A comanda contém pedidos . 3 - A comanda contém num. da mesa .
Classificação:	objeto
Impacto(s):	1 - O garçom joga a comanda. 2 - O caixa bota a comanda no escaninho . 3 - O caixa retira as comandas do escaninho . 4 - O caixa calcula o total das comandas 5 - O caixa anota o pedido na comanda, quando o cliente faz o pedido .
Sinônimo(s):	comandas.

Nome:	conta
Noção:	Valor total da despesa no restaurante . soma das opções consumidas incluido os 10%.
Classificação:	objeto
Impacto(s):	É conferido pelo caixa e é pago pelo cliente .
Sinônimo(s):	

Nome:	controle interno
Noção:	1 - Papel onde o caixa contém o total da despesa e o num. da mesa .
Classificação:	objeto
Impacto(s):	1 - O caixa anota o total e o num. da mesa na nota, quando fecha a mesa .
Sinônimo(s):	nota..

Nome:	divisão do escaninho.
Noção:	1 - É um quadrado do escaninho , usado para guardar as comandas referentes a uma das mesa . 2 - É identificada por um num. da mesa .
Classificação:	objeto
Impacto(s):	1 - O caixa pode botar a comanda na divisão do escaninho . 2 - O caixa pode retirar as comandas da divisão do escaninho . 3 - A divisão do escaninho pode estar vazia, mesa está fechada . 4 - A divisão do escaninho pode estar cheia, mesa está aberta . 5 - A divisão do escaninho pode contar comandas .
Sinônimo(s):	divisões..

Nome:	escaninho
Noção:	1 - Um dos componentes do restaurante . 2 - Contém várias divisões.
Classificação:	objeto
Impacto(s):	1 - O caixa bota a comanda no escaninho, quando recebe a comanda . 2 - O caixa retira as comandas do escaninho, quando fecha a mesa .
Sinônimo(s):	

Nome:	mesa
Noção:	1 - Um dos componentes do restaurante . 2 - Cada mesa tem um num. da mesa .
Classificação:	objeto
Impacto(s):	1 - O cliente pode sentar a mesa . 2 - O cliente pode trocar de mesa . 3 - A mesa está aberta ou mesa está fechada .
Sinônimo(s):	mesas..

Nome:	num. da mesa
Noção:	1 - número que identifica a mesa . 2 - número que identifica uma divisão do escaninho .
Classificação:	objeto
Impacto(s):	1 - O garçom anota o num. da mesa na comanda . 2 - O caixa verifica o num. da mesa da comanda .
Sinônimo(s):	

Nome:	num. do carimbo
Noção:	1 - Identifica cada uma das linhas do bordero .
Classificação:	objeto
Impacto(s):	1 - O caixa anota o num. do carimbo na nota, quando fecha a mesa .
Sinônimo(s):	

Nome:	opção
Noção:	1 - É um dos pratos ou bebidas do restaurante . 2 - Toda opção tem preço .
Classificação:	objeto
Impacto(s):	1 - O cliente escolhe opções, quando faz o pedido .
Sinônimo(s):	opções..

Nome:	pagamento
Noção:	1 - Dinheiro entregue pelo cliente , quando o cliente paga a conta .
Classificação:	objeto
Impacto(s):	1 - O caixa deve conferir o pagamento.
Sinônimo(s):	dinheiro..

Nome:	pedido
Noção:	1 - É uma opção . 2 - Tem um cliente .
Classificação:	objeto
Impacto(s):	1 - O garçom anota o pedido na comanda , quando o cliente faz o pedido.
Sinônimo(s):	pedidos..

Nome:	preço
Noção:	1 - Valor associado a opção .
Classificação:	objeto
Impacto(s):	1 - O caixa consulta preço quando calcula o total .
Sinônimo(s):	

Nome:	restaurante
Noção:	1 - É um lugar onde trabalham garçons e caixa . 2 - É um lugar onde são atendidos clientes . 3 - Tem opções para refeição. 4 - Tem mesas . 5 - Tem escaninho . 6 - Tem controle de caixa .
Classificação:	objeto
Impacto(s):	1 - O restaurante pode ter 10%. 2 - O restaurante pode não ter 10%. 3 - O cliente pode sentar a mesa do restaurante. 4 - O cliente pode trocar de mesa no restaurante. 5 - O cliente pode pedir a conta no restaurante. 6 - O cliente pode pagar a conta no restaurante.
Sinônimo(s):	

Nome:	total
Noção:	1 - Somatório do preço de cada pedido do cliente . 2 - Inclui os 10% , se o restaurante tem 10% .
Classificação:	objeto
Impacto(s):	1 - O caixa anota o total na nota, quando fecha a mesa . 2 - Quantia que devera ser paga no caixa .
Sinônimo(s):	soma.

Nome:	troco
Noção:	1 - Troco = pagamento - total .
Classificação:	objeto
Impacto(s):	1 - O caixa calcula o troco quando confere o pagamento .
Sinônimo(s):	

Símbolos do Tipo Verbo

Nome:	conferir pagamento.
Noção:	1 - Tarefa realizada pelo caixa . 2 - O caixa compara o pagamento com o total da nota (ou controle interno).
Classificação:	verbo
Impacto(s):	1 - Se pagamento for maior ou igual ao total , o caixa deve entregar o troco e o canhoto para o garçon . 2 - Se pagamento for menor que total , o caixa diz ao garçon que o pagamento está errado.
Sinônimo:	confere o pagamento, confere, conferir o pagamento,

Nome:	abrir a mesa.
Noção:	1 - Tarefa realizada pelo caixa . 2 - Acontece quando o cliente senta na mesa e faz o pedido . 3 - O caixa verifica se a mesa está fechada. 4 - O caixa recebe a comanda e bota a comanda no escaninho.
Classificação:	verbo
Impacto(s):	1 - A mesa está aberta. 2 - Se a mesa está aberta, o caixa não pode abrir a mesa , caixa diz para o garçon : mesa está aberta.
Sinônimo(s):	

Nome:	botar a comanda no escaninho
Noção:	1 - tarefa realizada pelo caixa . 2 - acontece quando o caixa recebe a comanda . 3 - o caixa verifica o num. da mesa da comanda . 4 - o caixa coloca a comanda na divisão do escaninho correspondente ao num. da mesa .
Classificação:	verbo
Impacto(s):	1 - Divisão do escaninho recebe nova comanda .
Sinônimo(s):	botar a comanda,, bota a comanda no escaninho..

Nome:	calcular o total
Noção:	1 - tarefa realizada pelo caixa . 2 - acontece quando o caixa fecha a mesa . 3 - para cada comanda , o caixa verifica o preço dos pedidos e adiciona a soma . 4 - se o restaurante cobra 10% , o caixa calcula os 10% e adiciona a soma .
Classificação:	verbo
Impacto(s):	1 - o caixa tem a nota.
Sinônimo(s):	calcula o total,, calcula a soma,.

Nome:	conferir pagamento.
Noção:	1 - Tarefa realizada pelo caixa . 2 - O caixa compara o pagamento com o total da nota (ou controle interno).
Classificação:	verbo
Impacto(s):	1 - Se pagamento for maior ou igual ao total , o caixa deve entregar o troco e o canhoto para o garçom . 2 - Se pagamento for menor que total , o caixa diz ao garçom que o pagamento está errado.
Sinônimo(s):	conferir o pagamento,, confere,, confere o pagamento..

Nome:	controle de caixa
Noção:	1 - É feito pelo caixa . 2 - Compreende bordero , escaninho , comandas .
Classificação:	verbo
Impacto(s):	1 - É fiscalizado pelo gerente.
Sinônimo(s):	

Nome:	entregar o pagamento.
Noção:	1 - Tarefa realizada pelo garçom . 2 - O garçom apresenta a nota ao cliente e recebe pagamento . 3 - O garçom devolve a nota ao caixa . 4 - O garçom entrega o dinheiro ao caixa .
Classificação:	verbo
Impacto(s):	1 - O caixa deve conferir o pagamento .
Sinônimo(s):	paga no caixa..

Nome:	entregar o troco
Noção:	1 - O caixa entrega o troco ao garçom, quando o caixa confere o pagamento .
Classificação:	verbo
Impacto(s):	1 - O garçom deve levar o troco ao cliente .
Sinônimo(s):	

Nome:	fazer o pedido
Noção:	1 - Ação realiada pelo cliente . 2 - O cliente escolhe opções do restaurante .
Classificação:	verbo
Impacto(s):	1 - O garçom anota as opções do cliente na comanda . 2 - O garçom deve jogar a comanda .
Sinônimo(s):	faz o pedido..

Nome:	fechar a mesa
Noção:	1 - Tarefa realizada pelo caixa . 2 - Acontece quando o cliente pede a conta . 3 - O caixa verifica o num. da mesa . 4 - O caixa verifica se a mesa esta aberta. 5 - O caixa retira as comandas do escaninho . 6 - O caixa calcula a soma . 7 - O caixa anota o total no controle interno . 8 - O caixa granpeia a nota (ou controle interno) com as comandas . 9 - O caixa carimba a nota com o num. do carimbo corrente no bordero . 10 - O caixa anota o num. da mesa e a quantidade de comandas no bordero . 11 - O caixa anota os 10% no canhoto .
Classificação:	verbo
Impacto(s):	1 - A mesa está fechada. 2 - O garçom deve entregar o pagamento ao caixa . 3 - Se a mesa está fechada, o caixa não pode fechar a mesa , caixa diz para o garçom : mesa está fechada.
Sinônimo(s):	fecha mesa,, fechou a mesa,, fecha a mesa..

Nome:	jogar a comanda
Noção:	1 - Tarefa realizada pelo garçom . 2 - O garçom coloca a comanda no balcão. 3 - Acontece quando o cliente faz o pedido .
Classificação:	verbo
Impacto(s):	1 - O caixa deve botar a comanda no escaninho .
Sinônimo(s):	recebe a comanda,, joga o comanda,,

Nome:	pagar a conta
Noção:	1 - Ação realizada pelo cliente . 2 - O cliente entrega o dinheiro ao garçon.
Classificação:	verbo
Impacto(s):	1 - O garçom deve entregar o pagamento ao caixa .
Sinônimo(s):	paga a conta,, recebe pagamento,, receber o pagamento,,

Nome:	pedir a conta
Noção:	1 - Ação realizada pelo cliente . 2 - Acontece quando o cliente não tem mais pedidos.
Classificação:	verbo
Impacto(s):	1 - O caixa deve fechar a mesa .
Sinônimo(s):	pedir a nota,, pede a conta..

Nome:	retirar comandas do escaninho
Noção:	1 - Tarefa realizada pelo caixa . 2 - O caixa pega as comandas da divisão do escaninho correspondente ao num. da mesa .
Classificação:	verbo
Impacto(s):	1 - A divisão do escaninho está vazia. 2 - O caixa deve calcular o total .
Sinónimo(s):	retirar as comandas,, retira as comandas.

Nome:	sentar a mesa
Noção:	1 - Ação realizada pelo freguês. 2 - Acontece quando o freguês escolhe uma das mesas.
Classificação:	verbo
Impacto(s):	1 - O freguês deve fazer o pedido ao garçom .
Sinónimo(s):	senta a mesa.

Nome:	transferir de mesa para mesa
Noção:	1 - Tarefa realizada pelo caixa . 2 - Acontece quando o freguês troca de mesa (origem) para a mesa (destino). 3 - O caixa verifica se a mesa está fechada. 4 - O caixa transfere as comandas da divisão do escaninho correspondente ao num. da mesa (origem) para a divisão do escaninho correspondente ao num. da mesa (destino).
Classificação:	verbo
Impacto(s):	1 - A divisão do escaninho correspondente ao num. da mesa (origem) está vazia. 2 - A mesa está fechada (origem). 3 - A divisão do escaninho correspondente ao num. da mesa (destino) recebe as comandas . 4 - A mesa está aberta (destino).
Sinónimo(s):	transferiu a mesa,, transfere mesa,.

Nome:	trocar de mesa
Noção:	1 - Ação realizada pelo cliente . 2 - Acontece quando o cliente muda de uma mesa para outra mesa .
Classificação:	verbo
Impacto(s):	1 - O caixa deve transferir de mesa para mesa .
Sinónimo(s):	troca,.

Símbolos do Tipo Estado

Nome:	mesa está aberta
Noção:	1 - É uma mesa . 2 - A mesa está aberta, se o caixa abriu a mesa e o caixa ainda não fechou a mesa , nem transferiu a mesa . 3 - A divisão do escaninho corresponde ao num. da mesa contém comandas (não está vazia) 4 - A mesa tem clientes.
Classificação:	estado
Impacto(s):	1 - O caixa pode fechar a mesa . 2 - O caixa pode transferir de mesa para mesa . 3 - O caixa não pode abrir a mesa .
Sinónimo:	

Nome:	mesa está fechada
Noção:	1 - A mesa está fechada, se o caixa fechou a mesa e ainda não abriu a mesa . 2 - A mesa está fechada, se o caixa transferiu a mesa e ainda não abriu a mesa . 3 - A divisão do escaninho corresponde ao num. da mesa não contém comandas (está vazia)
Classificação:	estado
Impacto(s):	1 - O caixa pode abrir a mesa . 2 - O caixa não pode fechar a mesa . 3 - O caixa não pode transferir de mesa para mesa .
Sinónimo:	