

Ascânio Zago Júnior

Estratégia para revisão de casos de uso baseada em padrões

Monografia apresentada à Escola
Politécnica da Universidade de São
Paulo para obtenção do título de MBA
em Engenharia de Software.

Área de Concentração:
Engenharia de Software

Orientadora:
Prof^a. Maria Alice G. V. Ferreira

São Paulo
2005

MBA/ES

2005

Z13e

DEDALUS - Acervo - EPEL



31500020062

FICHA CATALOGRÁFICA

M 2005BP

Zago Jr. , Ascânio

Estratégia para revisão de casos de uso baseada em padrões

São Paulo, 2005, 100 p.

Dissertação(MBA) – Escola Politécnica da Universidade de São Paulo

1825987

OK

1

**Dedico esse trabalho a minha esposa
Eliana por toda a compreensão e apoio
para que eu pudesse, finalmente,
concluir com sucesso esse enorme
desafio.**

AGRADECIMENTOS

A professora e orientadora Maria Alice por toda a paciência e ajuda no desenvolvimento desse trabalho.

A professora e coordenadora do curso Selma Shin Shimizu Melnikoff pela atenção e gentileza prestada por todo transcorrer do curso

Aos demais professores que contribuíram diretamente com minha formação teórica que foi essencial para o desenvolvimento da monografia.

Aos meus pais, que apesar de todas as dificuldades vividas nos últimos anos, sempre me deram muita força para que eu pudesse concluir esse trabalho.

RESUMO

A utilização de casos de uso tem crescido rapidamente como a principal técnica que oferece um meio sistemático e intuitivo para descrever processos de negócios e de captura de requisitos funcionais.

No desenvolvimento de software orientado a objetos, casos de uso direcionam todo o processo de desenvolvimento, auxiliando as atividades do projeto, que incluem a criação, verificação e validação (V & V) da arquitetura do sistema. Recentes estudos mostraram que, dentre os processos de V & V, a inspeção de software – que analisa e verifica as representações do sistema, dentre elas os diagramas de projeto – é uma das técnicas mais efetivas para a eliminação de defeitos do sistema. Assim, num processo de software orientado por casos de uso, a inspeção de casos de uso é fundamental para a captura erros que, detectados numa fase mais avançada do processo, redundariam em custos muito mais elevados.

Padrões de projeto (*design patterns*) são estruturas recorrentes, comuns a vários sistemas de software diferentes e que se mostraram eficientes, podendo ser reusadas em situações similares, em novos projetos. Já Antipadrões (*anti-patterns*), são defeitos freqüentemente encontrados nos projetos, que podem redundar em soluções desastrosas para o software.

A proposta desse trabalho é apresentar uma estratégia para a revisão dos casos de uso, utilizando padrões e antipadrões de caso de uso. A estratégia surgiu como resultado de um trabalho que teve por objetivo revisar um conjunto de modelos de casos de uso relacionados ao desenvolvimento de um produto de *Workflow*. A revisão utilizou padrões e antipadrões específicos para casos de uso, que estão discutidos ao longo do desenvolvimento do trabalho.

ABSTRACT

The deployment of use cases has been growing rapidly as the main technique that offers an intuitive and systematic mean to describe business processes and the capture of functional requirements.

Concerning the development of object-oriented system, use cases drive the whole development process providing additional help to the project activities, what include the creation, verification and validation (V & V) of the systems' architecture.

Recent studies have shown that amongst the V & V processes, the system inspection – what analyses and verifies the system representations, among them the diagrams of the project – is one of the most effective techniques to the elimination of system failures. Therefore, in a process of object-oriented system driven by use cases, the use cases inspection is essential to the capture of errors that, detected in a more advanced phase of the process, would incur in higher costs.

Design patterns are recurrent structures commonly shared by various distinctive systems, and they have proved themselves as being efficient, offering the opportunity of being reused in similar situations in new projects. Anti-patterns on its turn are small defects frequently found in projects that may redound in disastrous solutions to the system. The objective of this paper is to present a strategy for the review of the use cases, employing patterns and anti-patterns of the use cases. This strategy has emerged as the result of a work that had as its objective to review a group of use case models related to the development of a Workflow product. The review has employed specific patterns and anti-patterns that will be discussed as this paper unfolds.

SUMÁRIO

LISTA DE FIGURAS	i
LISTA DE TABELAS	iii
LISTA DE ABREVIATURAS E SIGLAS	iv
1 INTRODUÇÃO	1
1.1 Considerações iniciais	1
1.2 Motivação	3
1.3 Objetivos	4
1.4 Metodologia	4
1.5 Conteúdo da Dissertação	5
2 CONCEITUAÇÃO	6
2.1 Visão Geral de Casos de Uso	6
2.2 Atores	8
2.3 Descrição de Casos de Uso	8
2.4 Relacionamentos entre Casos de Uso	9
2.4.1 Inclusões (<i>Includes</i> ou <i>uses</i>)	9
2.4.2 Extensões (<i>Extends</i>)	12
2.4.3 Generalização e Especialização	14
2.5 Instâncias de Casos de Uso	16
2.6 Gerenciamento de Processos de Negócios	17
2.6.1 Visão Geral sobre Workflow	17
2.6.2 Controle de Roteamento e Acesso	18
2.7 Conclusões	18
3 PADRÕES PARA CASOS DE USO	19
3.1 Antipadrões (<i>Anti-patterns</i>)	19
3.1.1 Comunicação entre casos de uso	19
3.1.2 Decomposição Funcional	20
3.1.3 Micro Casos de Uso	21
3.1.4 Fluxos Alternativos como Extensões	22
3.1.5 Casos de Uso de Negócios	23
3.1.6 Mistura de níveis de abstração	24
3.1.7 Múltiplos valores de negócio	25
3.1.8 Níveis de Controle de Acesso como Atores	26
3.2 Padrões (<i>Patterns</i>)	27
3.3 Catálogo de padrões – Templates	29
3.3.1 Grupo – Múltiplos Atores	29
3.3.2 Grupo – CRUD	33
3.3.3 Grupo – Regras de Negócio	37
3.3.4 Grupo – Atributos Comuns	39
3.3.5 Grupo – Casos de Uso Extensos	45
3.3.6 Grupo – Visões Ortogonais	48
3.3.7 Grupo – Seqüência de Casos de Uso	51
3.4 Catálogo de padrões – Domínio Específico	55
3.4.1 Grupo – Meio Externo Passivo	55
3.4.2 Grupo – Controle de Acesso	57
3.5 Conclusões	63

4	ESTRATÉGIA PARA REVISÃO DE CASOS DE USO	64
4.1	Introdução	64
4.2	Estrutura	65
4.3	Desenvolvimento.....	68
4.3.1	Módulo: Estruturação.....	68
4.3.1.1	Glossário	68
4.3.1.2	Perfil dos Autores.....	69
4.3.1.3	Avaliação da Narrativa.....	69
4.3.1.4	<i>Templates</i>	71
4.3.2	Módulo: Atores	71
4.3.2.1	Dispositivos como Atores	71
4.3.2.2	Papéis para Múltiplos Atores	73
4.3.3	Módulo: Objetivo e Conteúdo.....	77
4.3.3.1	Múltiplos Objetivos.....	77
4.3.3.2	Casos de Uso de Negócio.....	78
4.3.3.3	Níveis de Abstração	79
4.3.3.4	Avaliação dos Nomes.....	81
4.3.4	Módulo: Agrupamentos e Relacionamentos	82
4.3.4.1	Operações CRUD.....	82
4.3.4.2	Reuso via inclusão.....	85
4.3.4.3	Decomposição Funcional	87
4.3.4.4	Inclusão × Extensão	88
4.3.5	Módulo: Sequenciamento.....	92
4.3.5.1	Avaliação de Pré-Condições	92
4.3.5.2	Outros Diagramas UML.....	93
4.3.6	Referências	94
4.3.6.1	Casos de Uso Extensos.....	95
4.3.6.2	Regras de Negócio	96
4.3.6.3	Aspectos não Funcionais.....	96
4.3.7	Conclusões	97
5	CONSIDERAÇÕES FINAIS	98
6	REFERÊNCIA BIBLIOGRÁFICA	99

LISTA DE FIGURAS

Fig. 2.1 – Esquematisação das dependências de artefatos entre atividades de análise e projeto	7
Fig. 2.2 – Diagrama de casos de uso (UML), destacando o caso de uso Ligação Local	7
Fig. 2.3 – Exemplo de Generalização e Especialização de atores para sistema aéreo ...	8
Fig. 2.4 – Caso de uso <i>Sacar Dinheiro</i> de uma ATM.....	10
Fig. 2.5 – Fluxo Básico do caso de uso <i>Sacar Dinheiro</i>	10
Fig. 2.6 – Fluxo Básico do caso de uso <i>Checar Senha Cliente ATM</i>	11
Fig. 2.7 – Caso de uso <i>Sacar Dinheiro</i> de uma ATM com Caso de Uso de Extensão <i>Logar Transação</i>	12
Fig. 2.8 – Caso de uso <i>Sacar Dinheiro</i> com Ponto de Extensão identificado	13
Fig. 2.9 – Fluxo Básico do caso de uso <i>Logar Transação</i> com referência ao Ponto de Extensão <i>Usuário Autenticado</i>	14
Fig. 2.10 – Exemplo do relacionamento de generalização e especialização.....	15
Fig. 2.12 – Notação UML para uma instância do caso de uso <i>Sacar Dinheiro</i>	16
Fig. 2.13 – Exemplo de Workflow – <i>Solicitação de Compra de Software</i>	17
Fig. 3.1 – Modelo que associa incorretamente dois casos de uso	19
Fig. 3.2 – Decomposição funcional de <i>Compilar Fonte</i>	20
Fig. 3.3 – Opções de um menu de programa.....	22
Fig. 3.4 – Atores modelando níveis de acesso	26
Fig. 3.5 – Padrões de casos de uso para a categoria <i>Templates</i>	27
Fig. 3.6 – Padrões de casos de uso para a categoria <i>Domínio Específico</i>	28
Fig. 4.1 – Estrutura de Módulos, Tópicos e Padrões da estratégia.	66
Fig. 4.2 – Caso de uso <i>Gerar Pasta via Dispositivo</i> , antes da revisão.....	72
Fig. 4.3 – Caso de uso <i>Gerar Pasta via Dispositivo</i> , depois da revisão	72
Fig. 4.4 – Modelo e fluxo básico para o caso de uso <i>Redirecionar Atividade</i> , antes da revisão	73
Fig. 4.5 – Modelo e fluxo básico para o caso de uso <i>Redirecionar Atividade</i> , depois da revisão	74
Fig. 4.6 – Modelo e fluxo básico para o caso de uso <i>Desativar Grupo</i> , antes da revisão	74
Fig. 4.7 – Modelo e fluxo básico para o caso de uso <i>Desativar Grupo</i> , depois da revisão	75
Fig. 4.8 – Modelo de casos de uso para controle de acesso a formulários no SGP, antes da revisão	76
Fig. 4.9 – Modelo de casos de uso para controle de acesso de formulários no SGP, depois da revisão	76
Fig. 4.10 – Modelo do caso de uso <i>Manutenção e Publicação de Formulário</i> (no Repositório do SGP), antes da revisão	77
Fig. 4.11 – Modelo do caso de uso <i>Manutenção e Publicação de Formulário</i> (no Repositório do SGP), depois da revisão.....	78
Fig. 4.12 – Modelo do caso de uso <i>Gerenciar Fluxo</i> , antes da revisão	78
Fig. 4.13 – Modelo do caso de uso <i>Gerenciar Fluxo</i> , depois da revisão	79
Fig. 4.14 – Modelo do caso de uso <i>Exportar Fluxo</i> , antes da revisão.	80
Fig. 4.15 – Modelo do caso de uso <i>Exportar Fluxo</i> , depois da revisão.	80
Fig. 4.16 – Exemplos de casos de uso com nomes que podem ser corrigidos.....	81

Fig. 4.17 – Modelo de Casos de Uso para gerenciamento de <i>scripts</i> , antes da revisão	83
Fig. 4.18 – Modelo de Caso de Uso e descrição para gerenciamento de <i>scripts</i> , depois da revisão	85
Fig. 4.19 – Caso de uso <i>Assinar Regra</i> , antes da revisão.....	86
Fig. 4.20 – Caso de uso <i>Assinar Regra</i> , depois da revisão.	87
Fig. 4.21 – Caso de uso <i>Validar Fluxo para Publicação</i> , antes da revisão	87
Fig. 4.22 – Caso de uso <i>Validar Fluxo para Publicação</i> , depois da revisão.	88
Fig. 4.23 – Caso de uso <i>Criar Grupo</i> , antes da revisão	89
Fig. 4.24 – Caso de uso <i>Criar Grupo</i> , depois da revisão	90
Fig. 4.25 – Caso de Uso <i>Login</i> utilizando o Antipadrão <i>Fluxo Alternativo como Extensão</i> , antes da revisão.....	91
Fig. 4.26 – Caso de Uso <i>Login</i> , depois da revisão.	91
Fig. 4.27 – Modelo de casos de uso que usam relacionamentos de inclusão indicando pré-condição, antes da revisão	92
Fig. 4.28 – Modelo de casos de uso que usavam relacionamentos de inclusão indicando pré-condição, depois da revisão	93
Fig. 4.29 – Máquina de estados para navegação de algumas telas do SGP	94
Fig. 4.30 – Caso de uso <i>Criar Regra</i> , antes da revisão.....	95
Fig. 4.31 – Modelo do caso de uso <i>Criar Regra</i> , depois da revisão	95
Fig. 4.32 – Fluxo básico do caso de uso <i>Criar Pasta</i> com detalhamento da interface do usuário	96
Fig. 4.33 – Fluxo básico do caso de uso <i>Criar Pasta</i> sem detalhamento de interface de usuário	97

LISTA DE TABELAS

Tabela 2.1 – Estruturação de caso de uso segundo o UP	9
Tabela 2.2 – Ponto de Extensão definido no caso de uso <i>Sacar Dinheiro</i>	13
Tabela 3.1 – Padrão <i>Papéis Diferentes</i>	29
Tabela 3.2 – Padrão <i>Papel Comum</i>	31
Tabela 3.3 – Padrão <i>CRUD Completa</i>	33
Tabela 3.4 – Padrão <i>CRUD Parcial</i>	36
Tabela 3.5 – Padrão <i>Definição Estática</i>	38
Tabela 3.6 – Padrão <i>Modificação Dinâmica</i>	39
Tabela 3.7 – Padrão <i>Reuso</i>	40
Tabela 3.8 – Padrão <i>Adição</i>	41
Tabela 3.9 – Padrão <i>Especialização</i>	42
Tabela 3.10 – Padrão <i>Reuso Interno</i>	45
Tabela 3.11 – Padrão <i>Seqüência Longa</i>	46
Tabela 3.12 – Padrão <i>Múltiplos Fluxos</i>	47
Tabela 3.13 – Padrão <i>Especialização</i>	48
Tabela 3.14 – Padrão <i>Descrição</i>	50
Tabela 3.15 – Padrão <i>Seqüência de casos de uso</i>	51
Tabela 3.16 – Padrão <i>Meio Externo Passivo</i>	55
Tabela 3.17 – Padrão <i>Checagem Embutida</i>	58
Tabela 3.18 – Padrão <i>Checagem Explícita</i>	59
Tabela 3.19 – Padrão <i>Unidades de Segurança Dinâmica</i>	59
Tabela 3.20 – Padrão <i>Atribuição Interna</i>	60
Tabela 3.21 – Exemplo para o grupo de padrões – <i>Controle de Acesso</i>	60
Tabela 4.1 – Descrição das Situações relacionadas aos Tópicos de revisão.....	67

LISTA DE ABREVIATURAS E SIGLAS

ATM	Automatic Teller Machines
BPM	Business Process Management
CRUD	Create Read Update Delete
OO	Orientação a Objetos
SGP	Sistema Gerenciador de Processos
UML	Unified Modeling Language
UP	Unified Process
WfMC	Workflow Management Coalition

1 INTRODUÇÃO

Este capítulo apresenta o contexto geral do trabalho, alguns conceitos relativos à Orientação a Objetos, as principais motivações, objetivos, metodologia e estrutura da monografia.

1.1 Considerações iniciais

A natureza dos negócios vem mudando rapidamente e os sinais dessa mudança estão por toda parte. Métodos clássicos de produção, que foram utilizados sem grandes alterações nas últimas décadas, agora estão sendo substituídos por novos métodos que se baseiam fundamentalmente na informação e na tecnologia. Nesse novo tempo, conhecido como *Era da Informação*, surgem, a cada dia, novas empresas, que rapidamente alcançam liderança de mercado em sua área de atuação.

Informação e tecnologia são fatores críticos de sucesso para qualquer empresa, entretanto não podem ser analisados isoladamente. A mudança vem ocorrendo em todo o sistema de negócios: na cultura organizacional, nos processos e nas pessoas. A informação, apoiada pela tecnologia, serve como base de sustentação para essa mudança.

Nesse cenário de competição acirrada, torna-se fundamental o uso de sistemas de informação de alta qualidade que possam suportar com maior precisão e confiabilidade os processos de negócios. A qualidade do software está fortemente ligada à qualidade dos processos utilizados na organização onde ele é produzido. Segundo Curtis (2000), empresas capazes de integrar, harmonizar e acelerar seus processos de desenvolvimento e manutenção de software, terão primazia do mercado.

Um processo de desenvolvimento de software trata dos métodos e tecnologias utilizadas para apoiar, avaliar e melhorar as atividades envolvidas na construção ou atualização de um sistema informatizado (LINDVALL, 2000). Pesquisas de novas técnicas na área de engenharia de software são intensificadas com o propósito de facilitar o tratamento e o entendimento dos complexos cenários de negócios que regem a nova economia global.

Para Jacobson et al. (1997), “a engenharia de software é um processo de construção de vários modelos relacionados entre si. Cada modelo define um aspecto do sistema e é descrito usando vários diagramas e documentos. Cada modelo é examinado ou manipulado por pessoas diferentes, com diferentes interesses específicos, papéis e tarefas”. Assim, definem *Stackholder* – interessado no sistema – como qualquer pessoa que está interessada em usar o sistema, entendê-lo, colocar requisitos para ele ou desenvolver uma ou mais partes dele. Um *Stackholder* pode ser um membro da empresa de desenvolvimento do software, como um engenheiro ou gerente, ou alguém de fora dela, como um usuário final, um instalador ou um comprador do

software. Cada um desses elementos pode enxergar um aspecto diferente do sistema, de forma que a equipe de desenvolvimento necessita construir vários modelos diferentes, um para cada interessado, mas todos eles correlacionados.

Os conceitos da Orientação a Objetos (OO) são considerados ideais para representar a maioria dos elementos que comparecem nesses diferentes modelos. Um Objeto é, para Jacobson et al. (1997), uma construção de modelagem e de programação que representa alguma “coisa” do mundo real e que comparece, também, nos modelos do sistema. Esta construção apresenta informação (dados, estados) e métodos (funcionamento). Visões distintas dos vários objetos e de seus relacionamentos são mostradas nos vários tipos de modelos criados ao longo do processo de desenvolvimento, de tal forma que cada visão apresenta algum aspecto de interesse do sistema.

Em cada estágio de desenvolvimento do software, é interessante elaborar modelos do sistema em diferentes níveis de detalhe, atendendo assim aos interesses particulares de cada *Stackholder*. Um dos primeiros modelos a ser desenvolvido é o Modelo de casos de uso. Um caso de uso especifica o comportamento de um sistema computacional – ou de uma de suas partes - e descreve uma seqüência de ações realizadas para produzir um resultado de valor para um usuário (BOOCH, RUMBAUGH, JACOBSON, 2000).

No desenvolvimento de software orientado a objetos, casos de uso direcionam todo o processo de desenvolvimento, auxiliando as atividades do projeto, que incluem a criação, verificação e validação (V & V) da arquitetura do sistema. Recentes estudos mostraram que, dentre os processos de V & V, a inspeção de software – que analisa e verifica as representações do sistema, dentre elas os diagramas de projeto – é uma das técnicas mais efetivas para a eliminação de defeitos do sistema (SOMMERVILLE, 2003). Assim, num processo de software orientado por casos de uso, a inspeção de casos de uso é fundamental para a captura erros que, detectados numa fase mais avançada do processo, redundariam em custos muito mais elevados.

Casos de uso são escritos em uma linguagem natural e fornecem um formato simples e estruturado, que possibilita o trabalho conjunto da equipe de desenvolvimento, bem como de todos os demais envolvidos no projeto.

A utilização de casos de uso pode trazer benefícios para todo o processo de desenvolvimento (ARMOUR-MILLER, 2003):

- Casos de uso são o veículo de captura de requisitos.
- Casos de uso são a base para definição dos requisitos funcionais.
- Casos de uso auxiliam na delimitação do sistema (fronteira).
- Casos de uso são um meio para comunicação com usuários finais e clientes.
- Casos de uso são a base para derivação de objetos.
- Casos de uso permitem fazer o mapeamento para os requisitos.
- Casos de uso são a base para a interface com o usuário.

- Casos de uso são o mecanismo para definir a interação entre objetos e suas interfaces.
- Casos de uso definem padrões de acesso para o banco de dados.
- Casos de uso são a base para o desenvolvimento incremental.
- Casos de uso consolidam um caminho padrão para representação de processos de negócios.
- Casos de uso auxiliam na estimativa do tamanho do projeto e nos recursos requeridos.
- Casos de uso fornecem a base para a documentação do sistema e manual do usuário.
- Casos de uso são uma ferramenta para controle de projeto.
- Casos de uso direcionam as atividades de desenvolvimento.
- Casos de uso definem casos de teste.

A extensa lista apresentada ressalta a importância desse modelo para o restante do processo de desenvolvimento de software.

1.2 Motivação

Um dos maiores desafios dentro de um processo de desenvolvimento de sistemas é assegurar que todos os envolvidos nesse processo tenham um entendimento único e correto do que precisa ser construído ou modificado, pois requisitos não identificados, incompletos ou que não reflitam as reais necessidades de negócio, trazem consequências graves que vão, desde o aumento nos custos e prazos, até o cancelamento total do projeto.

Estabelecer um mecanismo padrão que facilite a comunicação entre todos os participantes de um projeto é muito importante. Contudo, a criação desse mecanismo por si só não é suficiente; torna-se fundamental, utilizá-lo corretamente. Essa análise pode ser aplicada também para os casos de uso, ou seja, os casos de uso constituem uma técnica amplamente utilizada pela sua simplicidade, mas que na maioria das vezes é aplicada erroneamente, resultando, no fim, modelos confusos e incompletos.

Nesse contexto, a motivação principal dessa monografia é aprofundar o estudo sobre os casos de uso, analisando as principais falhas que acontecem durante o processo de modelagem, assim como analisar padrões para casos de uso (*use case patterns*), que visam aumentar o reuso de soluções de problemas recorrentes. O estudo visa a formação do embasamento teórico para a apresentação e o desenvolvimento da *Estratégia para Revisão de Casos de Uso*.

A estratégia surgiu como consequência do trabalho realizado em um projeto real dentro da empresa do autor da monografia, onde um dos sistemas desenvolvidos teve de ser reprojeto por não atender às características exigidas pelos *Stackholders*. A estratégia tem uma abordagem prática e focaliza a correção de problemas recorrentes a partir de soluções prontas e reutilizáveis.

Considerando que os casos de uso são o veículo de captura de requisitos, é fundamental que o desenho da solução e definição da arquitetura tenha como base modelos de casos de uso com alta qualidade. Dessa forma, dentro de um processo de desenvolvimento, a estratégia deve ser usada, antes das fases de análise e projeto. Isso significa que a estratégia pode ser usada durante toda a etapa de levantamento de requisitos, tanto na fase inicial de criação dos modelos, quanto na fase final com os modelos prontos, para efetuar uma revisão junto aos *Stackholders*.

1.3 Objetivos

O objetivo dessa monografia é apresentar a Estratégia para Revisão de Casos de Uso, estruturada a partir da experiência do autor durante um processo de desenvolvimento de um produto de *Workflow*. A revisão é baseada em padrões (*patterns*) que sintetizam as melhores práticas e refletem o conhecimento e experiência acumulada de vários profissionais que trabalham com modelagem de casos de uso.

Padrões de projeto (*design patterns*) são estruturas recorrentes, comuns a vários sistemas de software diferentes, e que se mostraram eficientes, podendo ser reusadas em situações similares, em novos projetos (GAMAA et al., 1995). Esta idéia, inicialmente proposta para a etapa de projeto do ciclo de vida de um sistema, pode ser aplicada também à construção dos casos de uso, reutilizando “componentes” de casos de uso (padrões). A identificação precoce destes componentes ou padrões permite que se reutilize mais facilmente, nas etapas que se seguem, estruturas de análise, projeto e codificação.

1.4 Metodologia

O desenvolvimento desse trabalho foi baseado em um modelo incremental contendo as seguintes fases que definiram a metodologia utilizada:

- 1- Pesquisa de Padrões: levantamento de trabalhos relacionados à padronização, reaproveitamento e qualidade de casos de uso. A pesquisa utilizou fontes como artigos, livros e publicações na Internet, de maneira geral.
- 2- Análise Conceitual: nessa fase, eram analisados os conceitos sobre casos de uso, necessários para entendimento dos padrões a serem descritos de acordo com os resultados da pesquisa da fase anterior.
- 3- Pesquisa de Falhas: etapa responsável pela pesquisa das falhas mais comuns – os antipadrões ou *anti-patterns* -, geradas durante a criação de modelos de casos de uso, e aplicação de possíveis padrões para correção dessas falhas.
- 4- Desenvolvimento: estruturação e desenvolvimento da estratégia de revisão de casos de uso, utilizando o embasamento teórico obtido nas fases anteriores.

1.5 Conteúdo da Dissertação

Esse trabalho está estruturado da maneira que se descreve em seguida.

O Capítulo 2 apresenta a base teórica sobre a modelagem com casos de uso e os principais conceitos da área de gerenciamento de processos com *Workflow*.

O Capítulo 3 apresenta um conjunto de padrões e antipadrões para casos de uso que foram utilizados para composição da Estratégia de Revisão de Casos de Uso. Conceituam-se padrões e antipadrões, antes da apresentação do conjunto.

O Capítulo 4 apresenta a estratégia de revisão de casos de uso, assim como sua aplicação em um sistema de *Workflow*. Os exemplos apresentados nessa parte foram retirados diretamente desse trabalho.

O Capítulo 5 apresenta as conclusões gerais da monografia e sugestão de novos trabalhos que poderão ser desenvolvidos, no futuro, relativos ao assunto aqui discutido.

2 CONCEITUAÇÃO

Esse capítulo fornece os principais conceitos sobre casos de uso, além dos conceitos sobre Workflow, fundamentais para o entendimento dos padrões de casos de uso e exemplos, que serão apresentados no próximo capítulo.

2.1 Visão Geral de Casos de Uso

No desenvolvimento Orientado a Objetos, os casos de uso constituem o modelo com menor nível de detalhamento, o mais próximo dos requisitos do sistema. A sua finalidade é definir o que um sistema deverá fazer, permitindo que implementadores e consumidores estabeleçam um acordo sobre sua essência. O modelo de casos de uso guia o desenvolvimento dos demais modelos do sistema. A Fig. 2.1 mostra a proposta do Método Unificado (UP), segundo Waslawicz (2004). Nessa figura estão esquematizadas as dependências de artefatos entre atividades de análise e projeto, onde os casos de uso se situam em posição privilegiada. Com isso se quer dizer que, sendo a atividade de projeto de casos de uso uma das primeiras no ciclo de vida de um software, eles deverão guiar o desenrolar das demais atividades. Também, falhas em seu projeto poderão implicar em custos elevados para correção nas etapas seguintes.

Casos de uso descrevem um conjunto seqüencial de ações, incluindo as variantes admitidas pelo sistema, para produzir um resultado observável e de valor para um ator (JACOBSON, 1999). Eles descrevem o que o sistema “faz” e não “como” isso é feito. Atores são os usuários dos serviços e podem ser pessoas, máquinas ou outros sistemas que interagem com o sistema considerado.

Casos de uso incluem as ações realizadas “dentro” do sistema, assim como a descrição dos eventos externos causados pelos atores para iniciar essas ações, mas não incluem ações realizadas “fora” do sistema. A descrição do fluxo de eventos deve incluir “como” e “quando” o caso de uso se inicia e termina, quando o caso de uso interage com os atores, além do detalhamento dos fluxos básicos e alternativos.

Todos os comportamentos contidos no sistema, ou seja, todos os caminhos possíveis em que o sistema pode ser usado são capturados pelos casos de uso. Casos de uso e Atores, assim como os relacionamentos entre eles, formam o Modelo de Casos de Uso (BITTNER e SPENCE, 2002).

Um Modelo de Casos de Uso pode ser representado por um diagrama UML conforme descrito em Booch et al. (1999). A notação de um caso de uso, em um diagrama UML, é uma elipse juntamente ao seu nome. Um ator, por sua vez, é representado por uma figura similar a uma pessoa. Os atores poderão estar conectados aos casos de uso através de Associações. Uma Associação indica que o caso de uso e o ator comunicam-se entre si, cada um com a possibilidade de enviar e

receber mensagens. A notação UML de uma Associação é uma linha contínua que liga o ator ao caso de uso.

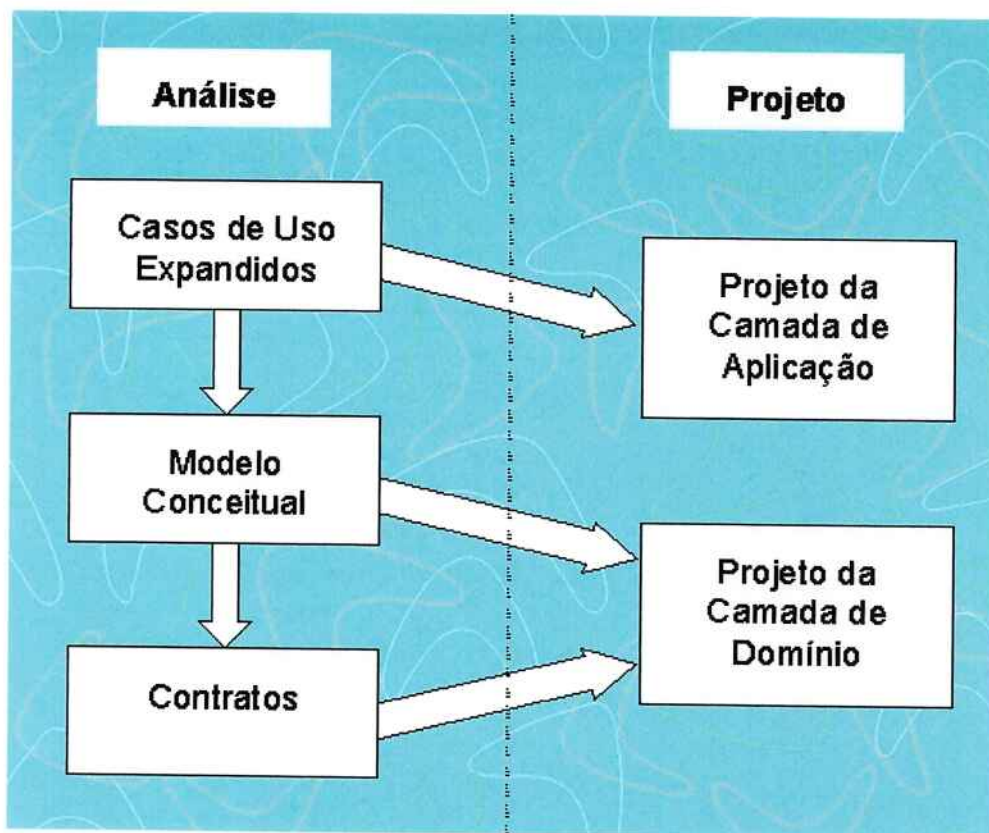


Fig. 2.1 – Esquematização das dependências de artefatos entre atividades de análise e projeto

A Fig. 2.2 mostra um exemplo de um diagrama UML para o caso de uso *Ligação Local*. O ator Assinante Chamador inicia o fluxo de eventos do caso de uso que tem como objetivo estabelecer uma chamada entre o primeiro ator citado e o ator Assinante Chamado.



Fig. 2.2 – Diagrama de casos de uso (UML), destacando o caso de uso Ligação Local

2.2 Atores

Segundo Kulak e Guiney (2003), um caso de uso nunca inicia um conjunto de ações por conta própria. O ponto de partida de qualquer interação é um ator.

Um ator é uma entidade que interage com o sistema com a proposta de completar um evento (JACOBSON, 1992). Tipicamente, um ator representa um papel que um ser humano, um dispositivo de hardware ou um outro sistema computacional desempenha com relação ao sistema.

A Generalização de atores é usada para expressar que um ator incorpora o papel definido por outro ator. Este é o único tipo de relacionamento que existe entre atores. O Ator que modela o papel comum é chamado de Ator Pai e aquele que incorpora o papel comum é chamado de Ator Filho (OVERGAARD e PALMKVIST, 2004). Esse tipo de relacionamento implica que o Ator Filho, além de interagir com os casos de uso que estão associados diretamente a ele, também interage com os casos de uso associados ao Ator Pai. Em outras palavras, um Ator Filho herda a capacidade de se comunicar com os casos de uso associados ao Ator Pai.

Suponha-se o exemplo da Fig. 2.3 que mostra um diagrama de casos de uso para um sistema de uma empresa de aviação civil. Os atores Balconista e Agente de Viagens podem solicitar uma reserva de passagem, mas do ponto de vista do caso de uso *Reservar Passagens*, existe somente uma entidade externa que realiza as interações em um dado momento. Dessa forma, o Balconista e o Agente de Viagens representam o mesmo papel para o caso de uso e podem ser modelados como Atores Filhos do ator Emissor de Reservas.

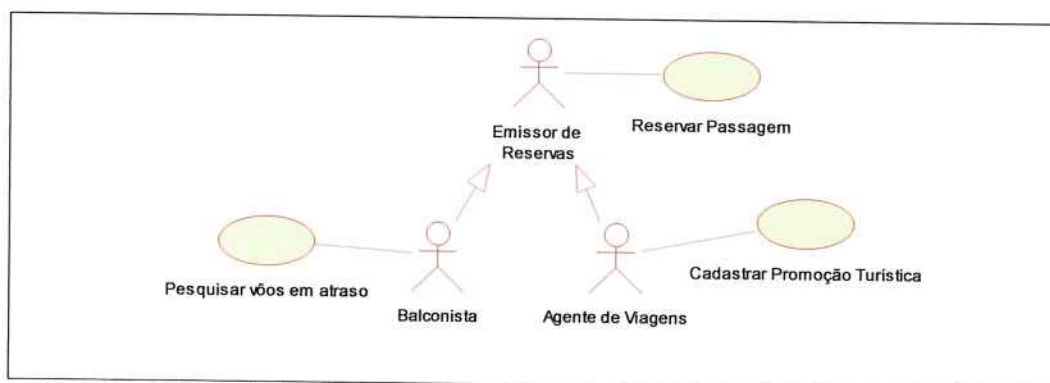


Fig. 2.3 – Exemplo de Generalização e Especialização de atores para sistema aéreo

2.3 Descrição de Casos de Uso

Um caso de uso é uma representação narrativa do comportamento do sistema. Ele é composto de um nome e uma descrição dos vários caminhos existentes para se atingir um objetivo determinado (ARMOUR e MILLER, 2003).

Descrever um caso de uso sem nenhum critério pode dificultar a leitura e gerar um entendimento errado do que o sistema deve fazer. Dessa forma, Jacobson (1992),

sugere a utilização de uma estruturação padrão para organizar a descrição de um caso de uso. Existem muitas variações e abordagens diferentes sobre como estruturar um caso de uso. A Tabela 2.1 mostra a estruturação padrão proposta pelo UP – Processo Unificado (RATIONAL, 2003). Cada linha da tabela pode ser considerada como uma seção da descrição do caso de uso.

Tabela 2.1 – Estruturação de caso de uso segundo o UP

Nome do caso de uso	Nome simples, que identifica o que o caso de uso descreve.
Descrição Resumida	Descrição breve da proposta do caso de uso. Normalmente, um parágrafo é suficiente para essa descrição.
Fluxo de Eventos	O Fluxo de eventos pode ser dividido em: – Fluxo Básico: descrição dos passos principais da seqüência de execução do caso de uso – Fluxos Alternativos: representação dos comportamentos alternativos ao Fluxo Básico de eventos. Geralmente, são descrições de exceções que podem acontecer no Fluxo Básico.
Requisitos Especiais	Um requisito especial normalmente representa um requisito não funcional, como por exemplo, detalhes de desempenho e usabilidade. Além disso, pode descrever restrições como ambiente de execução e compatibilidade entre sistemas.
Pré-Condição	Condição ou estado do sistema para que o caso de uso possa ser executado.
Pós-Condição	Uma lista de possíveis estados que o sistema pode apresentar logo que o caso de uso seja encerrado.
Pontos de Extensão	Descreve o nome e localização dos pontos de extensão, localizados no fluxo de evento. Pontos de extensão são definidos nesse capítulo na seção 2.4.2

2.4 Relacionamentos entre Casos de Uso

Nesse item serão apresentados os principais relacionamentos entre casos de uso: inclusões, extensões, generalizações e especializações, bem como o conceito de instância de casos de uso.

2.4.1 Inclusões (*Includes* ou *uses*)

No processo de identificação e descrição de casos de uso é comum observar um mesmo subfluxo de ações que se repete em dois ou mais casos de uso. Quando o subfluxo de ações pertence a casos de uso que possuem objetivos iguais e interagem com os mesmos atores, o mais adequado é juntar esses casos de uso em apenas um, ou utilizar um relacionamento de Generalização e Especialização. Entretanto, se o

subfluxo de ações for utilizado em casos de uso que têm propostas diferentes, deve-se avaliar a utilização desse tipo de relacionamento (inclusão).

Com um relacionamento de inclusão, é possível inserir dentro de um caso de uso, um fluxo descrito em outro caso de uso (BOOCH et al., 1999). Isso significa que o primeiro caso de uso, normalmente chamado de Caso de Uso Base, insere toda a seqüência de ações de um segundo caso de uso, Caso de Uso Incluído, como uma subseqüência.

A notação UML que representa o relacionamento de inclusão é uma linha pontilhada que se inicia no Caso de Uso Base e aponta para o Caso de Uso Incluído. Além disso, o esteriótipo <<includes>> deve ser utilizado junto ao relacionamento, conforme apresentado na Fig. 2.4. O caso de uso *Sacar Dinheiro* inclui todas as ações do caso de uso *Checar Senha Cliente ATM*.

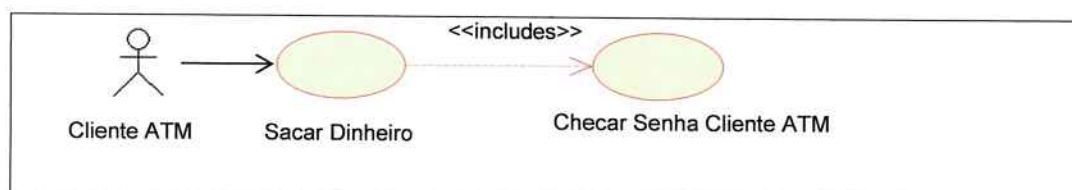


Fig. 2.4 – Caso de uso *Sacar Dinheiro* de uma ATM

A descrição do Caso de Uso Base de um relacionamento de inclusão deve conter referências explícitas sobre onde o Caso de Uso Incluído deverá ser colocado. A Fig. 2.5 mostra o Fluxo Básico de ações para o caso de uso *Sacar Dinheiro*, da Fig. 2.4. Note-se que esse fluxo está bastante voltado à implementação. Normalmente, recomenda-se que o caso de uso não entre em detalhes de implementação, sendo expresso em nível alto de abstração (WASLAWICZ, 2004). Entretanto, nesse exemplo, escolheu-se uma aplicação bem específica, o saque de dinheiro em uma caixa ATM.

Fluxo básico

O caso de uso se inicia quando o Cliente ATM insere o seu cartão magnético na leitora de cartões da ATM. O caso de uso lê as informações do cartão e solicita que a senha seja informada. O Cliente ATM informa a senha. **A senha é verificada de acordo com o Caso de Uso Incluído *Checar Senha Cliente ATM*.** O caso de uso armazena em um LOG¹ as informações do cartão e se a senha informada é válida. O caso de uso mostra os possíveis tipos de transações para o Cliente ATM.
[...]

Fig. 2.5 – Fluxo Básico do caso de uso *Sacar Dinheiro*

Um Caso de Uso Incluído é descrito como qualquer outro caso de uso. Entretanto, como esse tipo de caso de uso, freqüentemente, modela partes do uso e não o todo, ele é considerado um caso de uso Abstrato (OVERGAARD e PALMKVIST, 2004; JACOBSON et al., 1997). Isso significa que, não necessariamente, o caso de uso

¹ Todas as transações ocorridas na máquina ATM são armazenadas em um LOG de sistema.

deve iniciar a partir de um estímulo produzido por um ator ou deve gerar uma saída. Além disso, não é requerido que esse caso de uso possua um conjunto bem definido de ações de encerramento. Um Caso de Uso Incluído deve possuir uma seqüência contínua de ações, pois toda essa seqüência será inserida em um ponto do Caso de Uso Base.

Completando o exemplo da Fig. 2.5, o Fluxo Básico do caso de uso *Checar Senha Cliente ATM*, pode ser descrito no formato da Fig. 2.6.

Fluxo Básico

Quando o caso de uso é incluído por outros casos de uso, as seguintes ações são executadas:

[ações para verificação da senha]

O subfluxo termina e o processamento do caso de uso continua no Caso de Uso Base, posteriormente ao local onde esse caso de uso foi incluído. O resultado da verificação da senha é disponibilizado para o Caso de Uso Base.

Fig. 2.6 – Fluxo Básico do caso de uso *Checar Senha Cliente ATM*

Quando dois casos de uso estão conectados por um relacionamento de inclusão, o Caso de Uso Base é dependente do Caso de Uso Incluído, ou seja, o comportamento do Caso de Uso Incluído é inserido no comportamento do Caso de Uso Base. O Caso de Uso Incluído, por sua vez, não possui nenhum tipo de referência para o Caso de Uso Base podendo ser incluído em outros casos de uso.

2.4.2 Extensões (*Extends*)

Um relacionamento estendido entre casos de uso significa que o Caso de Uso Base incorpora, implicitamente, o comportamento de um outro caso de uso chamado de Caso de Uso de Extensão (BOOCH et al., 1999). Esse tipo de relacionamento é utilizado, principalmente, para modelagem da parte de um caso de uso em que o ator poderá considerar o uso de um comportamento opcional do sistema.

A notação UML que representa o relacionamento de extensão é uma linha pontilhada que inicia no Caso de Uso de Extensão e aponta para o caso de uso que utiliza a extensão (Caso de Uso Base). Além disso, o esteriótipo <<extends>> deve ser utilizado junto ao relacionamento.

Suponha-se o exemplo da Fig. 2.7. As ações relacionadas ao armazenamento no LOG, representado pelo caso de uso *Logar Transação*, podem ser consideradas uma extensão ao Fluxo Básico do caso de uso *Sacar Dinheiro*, podendo ser utilizada opcionalmente para algumas situações que aconteçam no fluxo. O ponto mais importante a ser considerado é que, adicionar as ações relacionadas ao LOG de transações como uma extensão, não muda o comportamento do caso de uso *Sacar Dinheiro*.

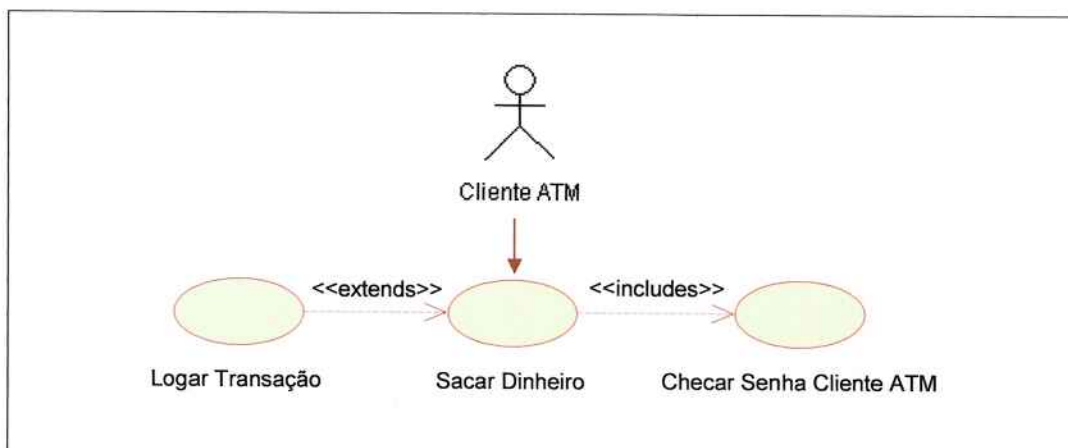


Fig. 2.7 – Caso de uso *Sacar Dinheiro* de uma ATM com Caso de Uso de Extensão *Logar Transação*

O Caso de Uso Base poderá permanecer isolado, mas sob certas condições, seu comportamento poderá ser estendido pelo comportamento de um outro caso de uso. O Caso de Uso Base poderá ser estendido somente em determinados pontos chamados Pontos de Extensão. Um Ponto de Extensão possui um nome e uma referência para o local da seqüência de ações em que pode ser estendido, conforme exemplo da Tabela 2.2.

Tabela 2.2 – Ponto de Extensão definido no caso de uso *Sacar Dinheiro*

Nome	Localização
Usuário Autenticado	Depois que a senha é verificada através do caso de uso <i>Checar Senha Cliente ATM</i> .

O nome do Ponto de Extensão deve resumir o que acontece nesse local da descrição do caso de uso. Esse nome não deve revelar o que será inserido nesse ponto para não criar uma dependência entre o Caso de Uso Base e o Caso de Uso de Extensão. Por exemplo, um nome errado seria: “Armazenar LOG da Autenticação”. Se futuramente, ao invés de armazenar um LOG nesse local, fosse necessário oferecer um novo serviço para o Cliente ATM, o nome do Ponto de Extensão perderia o sentido.

Conforme descrito em Overgaard e Palmkvist (2004), um relacionamento de extensão pode apresentar uma propriedade condicional que define se o Caso de Uso de Extensão será adicionado no Caso de Uso Base (no Ponto de Extensão indicado no próprio relacionamento). Considere que o caso de uso *Sacar Dinheiro* necessite armazenar um LOG de autenticação de um Cliente ATM apenas se esse processo ocorrer entre as 22 horas e as 06 horas. Em um diagrama UML, a propriedade condicional pode ser representada, de maneira descritiva, em um elemento de comentário conforme mostrado na Fig. 2.8.

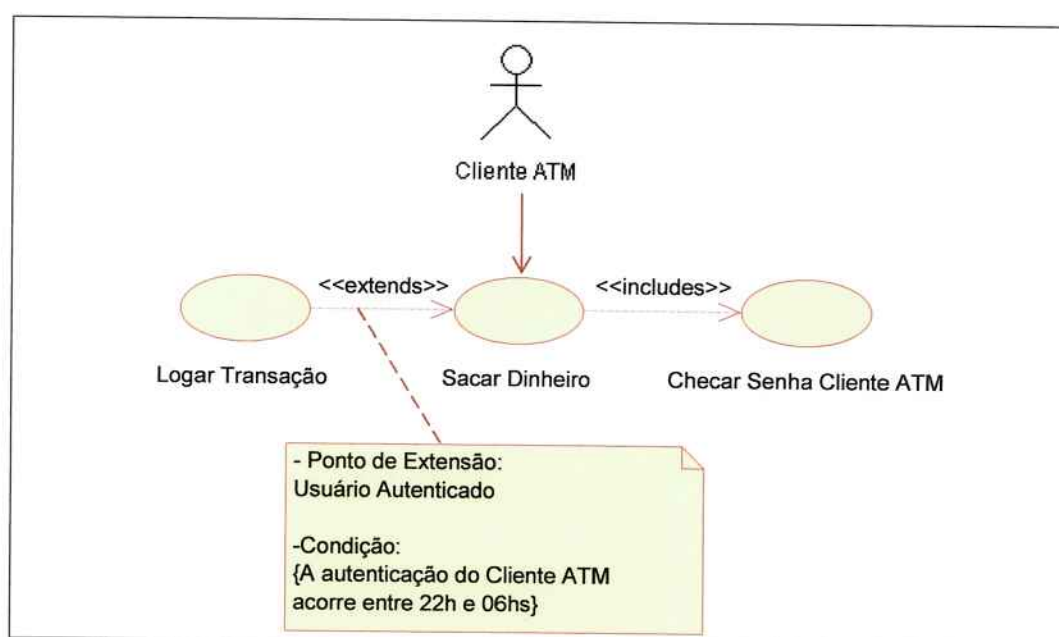


Fig. 2.8 – Caso de uso Sacar Dinheiro com Ponto de Extensão identificado

A descrição de um relacionamento de extensão é feita na própria descrição do fluxo do caso de uso de extensão, explicando como a inserção é iniciada. Essa descrição consiste de uma referência para o nome do Caso de Uso Base, uma referência para o

Ponto de Extensão onde o comportamento adicional será inserido e a condição para que o Caso de Uso de Extensão possa ser inserido. A Fig. 2.9 mostra a descrição do caso de uso *Logar Transação*.

Se o Caso de Uso de Extensão estende vários Casos de Uso Base, deverá ser criada uma descrição para cada Caso de Uso Base relacionado.

Fluxo Básico

O fluxo desse caso de uso é inserido dentro do caso de uso *Sacar Dinheiro*, no Ponto de Extensão *Usuário Autenticado*, se o horário de autenticação do Cliente ATM ocorre entre 22 e 06 horas.

[ações do LOG da transação]

O subfluxo termina e a instância do caso de uso continua no caso de uso *Sacar Dinheiro* depois do Ponto de Extensão *Usuário Autenticado*

Fig. 2.9 – Fluxo Básico do caso de uso *Logar Transação* com referência ao Ponto de Extensão *Usuário Autenticado*.

2.4.3 Generalização e Especialização

Em um Modelo de Casos de Uso é comum encontrar ações que modelam um mesmo tipo de uso do sistema. As ações comuns entre os casos de uso podem ser capturadas em um novo caso de uso genérico. Dessa forma, é possível criar relacionamentos de generalização e especialização entre os casos de uso.

O caso de uso que possui os atributos comuns (Generalização) é chamado de Caso de Uso Pai e os casos de uso especializados são chamados de Casos de Uso Filhos. Geralmente, os Casos de Uso Pai são abstratos, ou seja, não é possível instanciá-los; eles são utilizados apenas para capturar as ações comuns dos seus filhos.

A generalização entre dois casos de uso implica que todo o Caso de Uso Pai é herdado, isso significa que o Fluxo Básico com todos os Fluxos Alternativos e Pontos de Extensão devem ser herdados (ARMOUR e MILLER, 2003). Se somente uma parte deve ser reutilizada pelo Caso de Uso Filho, o relacionamento de generalização não é adequado. Para esse caso, deve-se pensar em separar apenas a parte comum em um novo caso de uso e utilizar relacionamentos de inclusão e extensão dos Casos de Uso Base para esse novo caso de uso.

A Fig. 2.10 mostra um diagrama UML para o caso de uso *Sacar Dinheiro*, com herança do caso de uso *Realizar Transação ATM*. A notação UML do relacionamento de generalização é uma linha contínua que sai do Caso de Uso Filho (especialização) e aponta para o Caso de Uso Pai (generalização).

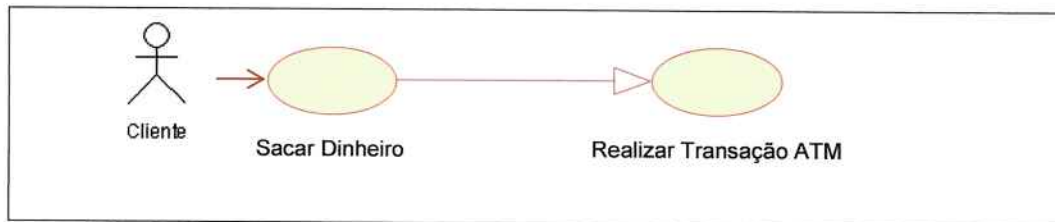


Fig. 2.10 – Exemplo do relacionamento de generalização e especialização

Na descrição dos Casos de Uso Filho devem existir referências para os Casos de Uso Pai. Essas referências implicam que o comportamento descrito no Caso de Uso Pai é executado no Caso de Uso Filho. Suponham-se as operações que podem ser realizadas em uma máquina ATM. Qualquer uma das operações tem um comportamento básico, que pode ser descrito como:

- O cliente entra com o cartão;
- O cliente informa uma senha;
- O cliente seleciona uma operação que é executada e encerrada;
- A ATM volta para seu estado inicial aguardando uma nova operação.

A Fig. 2.11 mostra a descrição para os casos de uso *Sacar dinheiro* e *Realizar Transação ATM* conforme proposto por Overgaard e Palmkvist (2004). O Fluxo Básico de ações para ambos os fluxos está dividido em três seções. O caso de uso *Sacar Dinheiro* herda as ações descritas nas seções *Início do Fluxo* e *Final do Fluxo* do Caso de Uso Pai e a seção *Execução da Transação* é especializada no caso de uso *Sacar Dinheiro*. A estrutura desse exemplo poderia suportar a descrição de outras operações, como por exemplo, *Gerar Extrato*, que também seria uma especialização de *Realizar Transação ATM*.

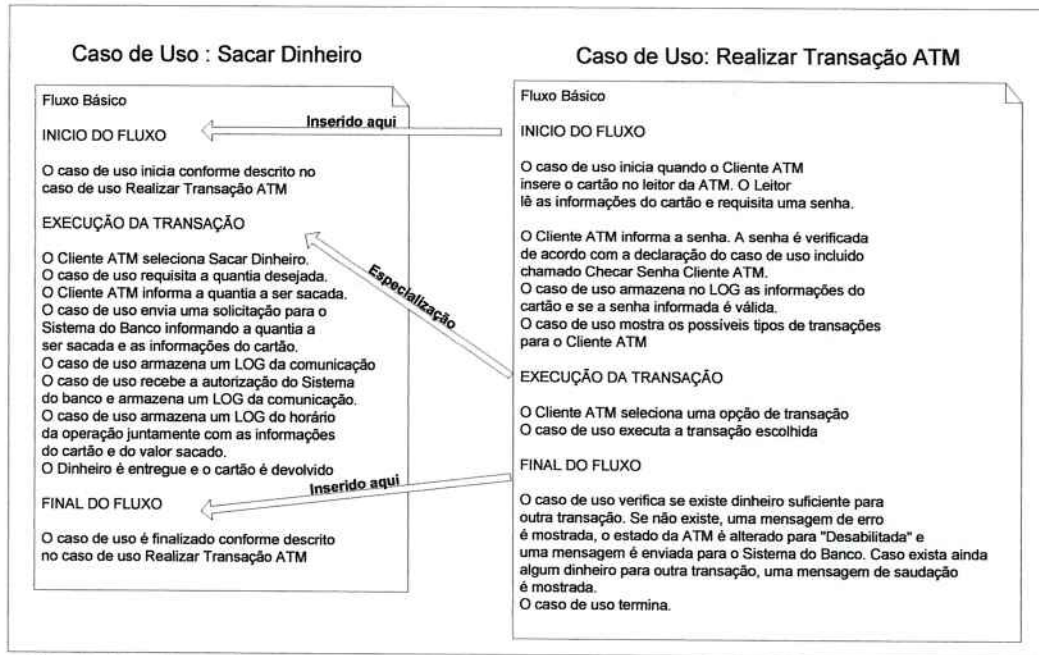


Fig. 2.11 – Descrição dos casos de uso Sacar Dinheiro e Realizar Transação ATM

2.5 Instâncias de Casos de Uso

Uma instância de um caso de uso modela a execução de um uso particular, seguindo um caminho dentre as seqüências de ações descritas por um caso de uso e iniciadas por um ator. Na UML, uma instância de caso de uso é chamada de Cenário (BOOCH et al., 1999).

Por exemplo, para o caso de uso *Sacar Dinheiro*, pode-se ter o seguinte cenário: “João saca R\$100,00 da sua conta-poupança, em uma máquina ATM, no dia 22 de Janeiro”. As instâncias são representadas graficamente, sublinhando-se seu nome, conforme mostrado na Fig. 2.12.

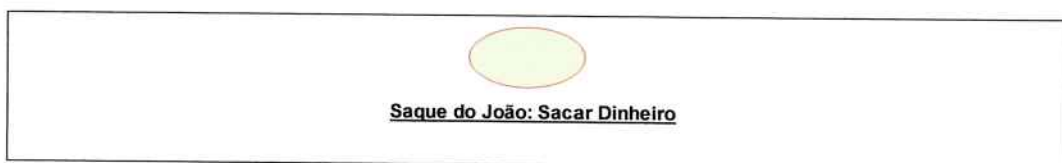


Fig. 2.12 – Notação UML para uma instância do caso de uso Sacar Dinheiro

A vantagem do conceito de instância de caso de uso é que possibilita a discussão de diferentes formas de se utilizar um caso de uso. Um caso de uso pode comportar-se de maneiras diferentes, dependendo do estado do sistema e das entradas fornecidas pelos atores.

2.6 Gerenciamento de Processos de Negócios

Nesse item serão apresentados os principais conceitos sobre a área de Gerenciamento de Processos de Negócios baseados em Workflow, visando o entendimento dos exemplos que serão apresentados no capítulo 4.

2.6.1 Visão Geral sobre Workflow

Segundo o WfMC (1999), Workflow é a automação de uma parte ou de todo o processo de negócio onde informações ou tarefas são passadas de um participante para outro, seguindo um conjunto de regras estabelecidas.

Um participante representa um recurso dentro do Workflow, podendo ser uma pessoa ou uma máquina. Os participantes realizam tarefas dentro de “passos” definidos dentro de um Workflow. Um passo representa uma unidade lógica de trabalho, que pode ser executada automaticamente por uma máquina ou, manualmente, por uma pessoa.

Os passos dentro de um Workflow estão conectados através de transições. Uma transição possui um passo de origem e um passo final. A Fig. 2.13 apresenta o Workflow *Solicitação de Compra de Software*, que exemplifica os conceitos apresentados até o momento. Solicitação, Aprovação, Compras, Recebimento, Instalação, Contabilização, Inventário e Avaliação Instalação representam os passos do fluxo de trabalho e as setas indicam as transições entre os passos. Alguns desses passos são manuais, como Solicitação e Instalação. Outros passos podem ser automáticos, como Contabilização e Inventário.

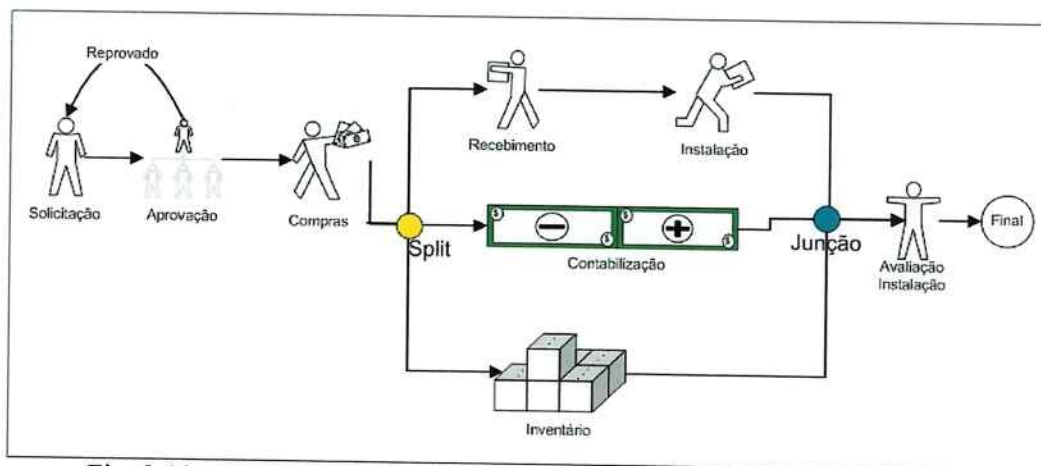


Fig. 2.13 – Exemplo de Workflow – *Solicitação de Compra de Software*

As tarefas em um Workflow podem ser executadas seqüencialmente e/ou paralelamente. No exemplo da Fig. 2.13, as tarefas associadas aos passos Solicitação, Aprovação e Compras são executadas seqüencialmente, ou seja, é necessário que as

tarefas de um passo finalizem para que o passo seguinte possa ser executado. As tarefas dos passos Recebimento/Instalação, Contabilização e Inventário são executadas paralelamente, ou seja, podem ser executadas em um mesmo momento uma vez que são independentes.

2.6.2 Controle de Roteamento e Acesso

As tarefas que devem ser executadas por cada passo estão agrupadas logicamente em pastas. Uma pasta contém uma lista de itens de trabalho que um participante pode interagir. O tipo mais comum de item de trabalho é o formulário.

A movimentação de uma pasta dentro de um Workflow pode ser chamada de roteamento. Conforme descrito em 2.6.1, o roteamento de uma pasta pode acontecer seqüencialmente ou paralelamente. Para que uma pasta possa seguir um caminho paralelo dentro de um fluxo, é necessário que exista um *split* entre os passos.

Um *split* é um ponto dentro do Workflow que divide o controle da pasta em dois ou mais roteamentos paralelos. Isso significa que a mesma pasta pode estar em dois ou mais passos diferentes dentro do fluxo. Da mesma forma, para que o controle da pasta volte para um roteamento seqüencial, é necessário utilizar uma junção (*joint*). Na Fig. 2.13, apresentada anteriormente, após o passo Compras, existe um *split* que cria um roteamento paralelo para três passos distintos. Uma junção pode ser assíncrona, ou seja, as tarefas nos passos independentes podem ser concluídas em momentos diferentes, antes que o roteamento passe a ser seqüencial novamente.

Uma variação do *split* é a ramificação. A ramificação é um ponto do Workflow onde é possível tomar uma decisão sobre que ramo do Workflow uma pasta deverá seguir. Nesse caso, o roteamento continua seqüencial e não há divisão do controle.

Cada passo dentro de um Workflow possui um conjunto de participantes da equipe de projeto que podem ter algum tipo de acesso à pasta. Basicamente, o controle de acesso define quais subgrupos podem ter acesso a uma pasta em um dado passo e quais os itens de trabalho dentro da pasta que os grupos poderão manipular.

O controle de acesso pode definir quem visualiza (controle de leitura) e quem altera (controle de edição) itens de trabalho dentro da pasta. O controle de acesso deve ser definido juntamente com a configuração do Workflow.

2.7 Conclusões

Os conceitos relativos aos casos de uso detalham dois pontos cruciais na criação dos modelos. O primeiro é identificar quem utiliza o sistema e quais os papéis que esses usuários representam para o sistema. O segundo ponto é observar o que o sistema deve fazer para esses usuários, em ordem, gerando um resultado de valor.

3 PADRÕES PARA CASOS DE USO

Esse capítulo apresenta um catálogo de antipadrões (*anti-patterns*) e padrões (*patterns*) de casos de uso, que foram utilizados como referência na montagem da Estratégia para Revisão de casos de uso, bem como o estudo de caso do sistema de *Workflow*.

3.1 Antipadrões (*Anti-patterns*)

Antipadrões são soluções ruins, que ao serem adotadas acarretam uma série de problemas. Normalmente, considera-se que antipadrões constituem uma extensão aos padrões. Os antipadrões são também soluções encontradas em modelagens reais de sistemas, porém elas trazem prejuízos ao modelo, ao invés de trazerem benefícios.

A identificação de antipadrões pode ser muito valiosa durante uma revisão dos casos de uso de um sistema. A identificação imediata de um antipadrão permite que se corrija facilmente a modelagem, empregando-se a solução recomendada para o problema detectado.

A seguir, listam-se alguns dentre os antipadrões mais comuns relatados.

3.1.1 Comunicação entre casos de uso

“Casos de uso modelam um uso completo do sistema.” (JACOBSON, 1992). Isso significa que uma instância de um caso de uso contém a execução completa de um uso do sistema, ou seja, a execução das ações dentro do sistema, assim como suas interações com pessoas e sistemas externos (atores).

Uma instância de um caso de uso nunca irá trocar mensagens com outras instâncias de casos de uso dentro de um mesmo sistema. A Fig. 3.1 apresenta dois casos de uso trocando mensagens. Se isso acontece, significa que a instância não executa um uso completo do sistema, pois ações adicionais são realizadas por outra instância. Pode-se entender que, se ações de uma outra instância precisam ser executadas para considerar um uso completo do sistema, o caso de uso associado a essa instância deve ser incluído no primeiro caso de uso. De outra forma, o primeiro caso de uso estará incompleto.

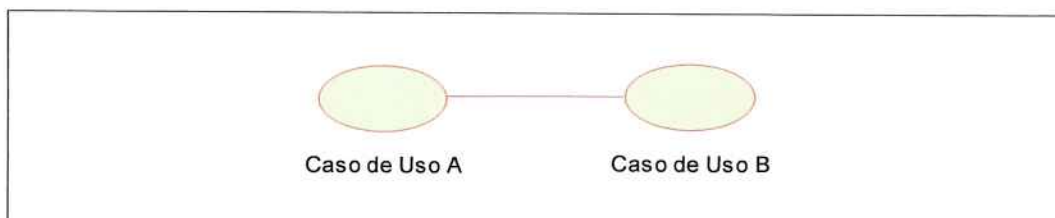


Fig. 3.1 – Modelo que associa incorretamente dois casos de uso

É comum observar comunicação entre casos de uso em modelos que foram gerados por desenvolvedores que tentam expressar a estrutura interna do sistema: modela-se o comportamento de uma parte do sistema em um caso de uso e outra parte do comportamento em outro caso de uso. Essa estruturação é errada, pois modelos de casos de uso modelam, por definição, o comportamento do sistema como um todo, e não suas partes. Além disso, modelos de casos de uso não devem expressar qualquer característica da estrutura interna do sistema.

A correção de modelos com esse tipo de falha é simples. Os dois casos de uso devem ser concatenados em apenas um. Os casos de uso originais e seus relacionamentos devem ser apagados do Diagrama de Casos de Uso. Se existe a necessidade de manter os dois casos de uso em separado no modelo, pode-se definir relacionamentos do tipo inclusão ou extensão entre eles, mantendo o objetivo primário que é existir apenas uma instância de caso de uso.

3.1.2 Decomposição Funcional

Técnicas tradicionais para decomposição de uma função em subfunções menores são usadas, por engano, para extração de subfluxos de um caso de uso. Esse subfluxos são organizados em novos casos de uso do tipo inclusão (*Include*). Dessa forma, o Caso de Uso Base pode ser visto como uma composição de Casos de Uso Incluídos.

Suponha-se a decomposição funcional de um compilador qualquer em subfunções: *Verificar Regras Gramaticais*, *Realizar Checagem Semântica*, *Realizar Checagem de Tipos* e *Gerar Código de Saída*. Criar um caso de uso para cada uma dessas subfunções e incluí-los através de relacionamentos de inclusão em um novo caso de uso chamado *Compilar Fonte* representa um exemplo típico de falha de modelagem. Basicamente, o caso de uso *Compilar Fonte* não conterá ações adicionais daquelas incluídas pelos outros casos de uso. A Fig. 3.2 mostra a decomposição desse exemplo.

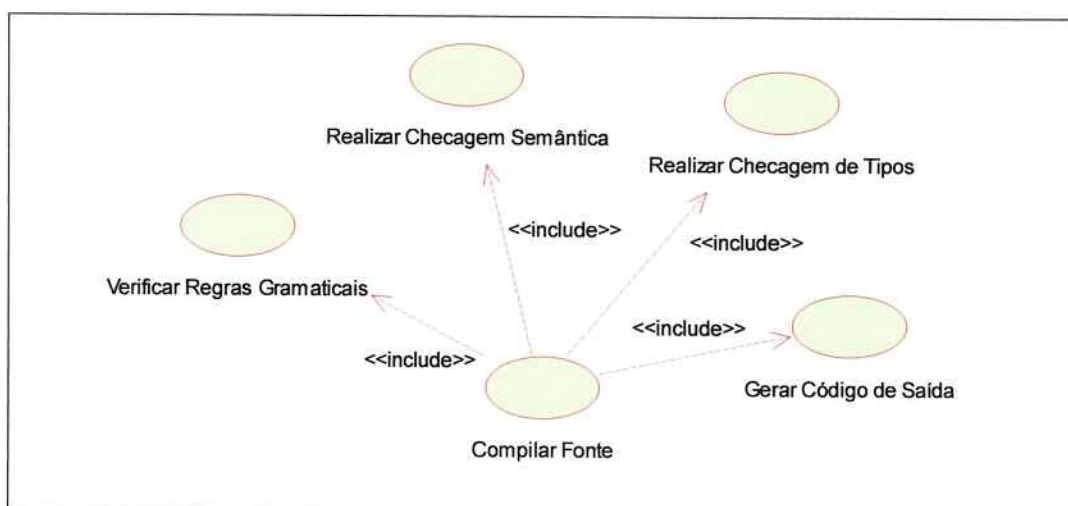


Fig. 3.2 – Decomposição funcional de *Compilar Fonte*

Casos de uso modelam usos do sistema. Cada um desses usos fornece um valor para um *Stakeholder* do sistema. Para tornar um modelo de casos de uso simples de ser mantido, é possível extrair subfluxos de um Caso de Uso Base, inserindo-os em casos de uso abstratos, que podem incluir ou estender as ações do Caso de Uso Base. Essa extração de subfluxos pode ser feita quando se deseja modelar atributos comuns entre casos de uso, quando existem diferentes configurações de sistemas, quando subfluxos estão definidos em diferentes camadas ou quando um *Stakeholder* requer que o subfluxo seja indicado explicitamente no modelo. Contudo, extrair subfluxos para dentro de casos de uso separados, simplesmente para indicar a divisão de módulos para os desenvolvedores não é aconselhável. O tamanho do modelo irá crescer, existirão mais documentos para se manter e certamente, da perspectiva de modelagem, a visão geral do uso completo do sistema estará perdida.

A correção de modelos que apresentam esse tipo de falha exige alguns passos. Inicialmente, deve-se identificar todos os casos de uso que realmente representam usos completos do sistema iniciados por atores. Em seguida, é necessário eliminar todos os casos de uso que não geram valor para um *Stakeholder*. Nesse caso, deve-se substituir a referência de inclusão no fluxo do Caso de Uso Base pelo fluxo do Caso de Uso Incluído. Posteriormente, os Casos de Uso Incluídos e seus relacionamentos podem ser removidos do modelo.

A tentativa de utilizar técnicas de decomposição funcional em casos de uso, normalmente indica que o sistema contém rotinas complexas e que é difícil criar modelos que, ao mesmo tempo, mostrem uma visão geral do sistema e apresentem detalhamento suficiente para o entendimento do uso do sistema. Nessas situações é fortemente aconselhável entender e aplicar um padrão de caso de uso que atenda essa situação, o *Caso de Uso Extenso*, que faz parte do catálogo descrito nesse trabalho.

3.1.3 Micro Casos de Uso

Muitos profissionais, ao modelarem um sistema, especialmente sistemas que já existem, descrevem as ações que os atores podem realizar no sistema focando na interface de usuário. Esse é o erro principal, pois a maioria esquece de perguntar “o quê” o usuário quer fazer com o sistema e focaliza-se em “como” o usuário deve interagir com o sistema do ponto de vista de telas.

Um exemplo de micro casos de uso seria um modelo onde cada entrada de um menu, em uma interface gráfica de usuário, é modelada como um caso de uso à parte. Nesse exemplo, o fluxo de cada caso de uso irá consistir de uma simples seleção executada pelo ator no menu e de uma resposta a essa ação. Na Fig. 3.3 é mostrado um exemplo dessa situação, um conjunto de opções de menus de um programa.

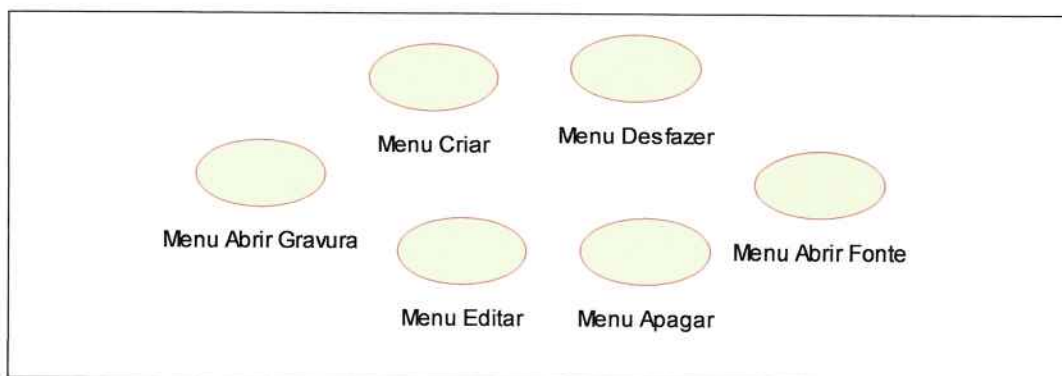


Fig. 3.3 – Opções de um menu de programa

Um bom indicador desse tipo de falha de modelagem é um número excessivo de casos de uso no modelo. Além disso, o estudo do nome dos casos de uso pode revelar que eles modelam operações muito simples e, por consequência, possuem a descrição do fluxo muito curta. Uma avaliação adicional que pode ajudar na detecção desse tipo falha é verificar se existe a necessidade de expressar a ordem em que o conjunto de casos de uso devem ser executados.

De maneira geral, uma informação simples, fornecida pelo usuário, e sua respectiva ação de resposta raramente fornece um valor de negócio para um *Stakeholder*, ou seja, nenhum uso do sistema será apenas fornecer um valor e receber a ação associada. Por essa razão, não há representação de uso completo do sistema.

Para corrigir modelos com esse tipo de falha, deve-se considerar como os usuários “gostariam” de usar o sistema. É necessário definir um caso de uso para cada conjunto de casos de uso que são executados juntos, ou seja, as seqüências que fazem um uso completo do sistema. Os novos casos de uso do modelo irão conter as ações dos micro casos de uso como parte do seus fluxos. Uma falha na correção desse modelo é criar relacionamentos de inclusão dos novos casos de uso para os micro casos de uso. Isso acaba gerando novamente um modelo incorreto, conforme discutido no item 3.1.2 (Decomposição Funcional).

3.1.4 Fluxos Alternativos como Extensões

Um caso de uso tem, geralmente, um único Fluxo Básico ou principal. Esses fluxos são executados quando “tudo vai bem”, ou seja, nenhuma situação anormal ou atípica foi encontrada durante a execução das ações. Entretanto, um caso de uso também pode incluir variações do Fluxo Básico para serem utilizadas se uma condição não é alcançada ou se algo de errado aconteceu. Por exemplo, um ator fornece, inadvertidamente, uma entrada não esperada. Essas variações que acontecem nos casos de uso são importantes da perspectiva do usuário, pois a não modelagem de qualquer uma delas pode resultar em um sistema que não funciona adequadamente. Do mesmo modo, adicionar um desses fluxos alternativos em outro fluxo pode gerar um comportamento não esperado do sistema e isto é um erro bastante comum, quando os profissionais que trabalham com modelagem extraem os fluxos alternativos para casos de uso do tipo Extensão.

Conforme mostrado no item 2.4.2, um relacionamento do tipo Extensão significa que o Caso de Uso de Extensão pode ser adicionado no fluxo do Caso de Uso Base. Isso implica que o comportamento do Caso de Uso Base não pode ser substituído ou modificado pelo caso de uso de extensão. Dessa forma, não é possível extrair um fluxo alternativo para dentro de um Caso de Uso de Extensão, pois o comportamento extraído será executado como uma adição ao que já existe no Caso de Uso Base e não uma substituição a uma determinada ação.

Basicamente, para corrigir modelos que têm esse tipo de falha, deve-se realizar o caminho inverso do que foi modelado, ou seja, capturam-se os fluxos dos Casos de Uso de Extensão e adiciona-se esses fluxos como fluxos alternativos nos Casos de Uso Base. Em seguida, deve-se apagar os Casos de Uso de Extensão e os relacionamentos associados.

Se, em um dado modelo, for altamente desejável que um fluxo alternativo seja expresso como um caso de uso a parte, pode-se utilizar um relacionamento de inclusão entre os dois casos de uso.

3.1.5 Casos de Uso de Negócios

De maneira geral, casos de uso são usados para modelar como o sistema é usado. Contudo, a mesma técnica é aplicável para modelar como uma empresa é usada por seu ambiente. Nesse contexto, casos de uso irão modelar diferentes processos, começando por alguém de “fora da empresa” como, por exemplo, um cliente; também serão modeladas as ações que serão realizadas “dentro da empresa” como respostas à interação da entidade externa. Esses casos de uso que modelam a empresa são chamados de Casos de Uso de Negócios. Fragmentos de Casos de Uso de Negócios podem ser realizados por Casos de Uso de Sistema, isto é, eles podem ser mapeados para os casos de uso tradicionais do modelo de Casos de Uso.

Quando um fluxo de negócio deve ser implementado por um sistema, não é óbvio deduzir se o fluxo será representado por um simples caso de uso no modelo de Casos de Uso de Sistema ou se será necessário criar vários Casos de Uso.

Um exemplo de Caso de Uso de Negócios é *Gerenciar Pedido* em um negócio de Comércio por Atacado. O fluxo contém, dentre outras ações: registrar o pedido, providenciar os itens para remessa, gerar o faturamento e registrar o pagamento do cliente. Além disso, pode-se pensar também nas ações de alteração e cancelamento do pedido. Mapear esse caso de uso para um Caso de Uso de Sistema é um erro, pois esse caso de uso não modela um uso do software, e sim um uso do negócio. O fluxo contém vários subfluxos que são realizados em diferentes momentos e, provavelmente, por diferentes pessoas. Cada um desses subfluxos fornece um valor para um *Stakeholder* inserido no negócio de Comércio por Atacado. Dessa maneira, cada subfluxo constitui um caso de uso no software.

O problema, ao incluir Casos de Uso de Negócios em um modelo de Casos de Uso de Sistema, é a mistura de propostas diferentes. Casos de Uso de Negócios,

normalmente, são escritos em um alto nível de abstração. Descrivê-los em um nível apropriado para Casos de Uso de Sistema certamente irá gerar um fluxo extenso e complexo.

De maneira geral, quando um caso de uso requer um ator para realizar alguma ação, como prover uma informação de entrada ou ser o receptor de uma informação de saída, isso implica em uma interrupção na execução do caso de uso. Se o dado requerido pelo caso de uso pode ser informado em um momento bastante posterior à última ação executada, pode-se entender que essa ação já não faz parte desse caso de uso. Dessa forma, um novo caso de uso deverá ser criado e seu início se dará pela introdução do dado requerido, que interrompeu o caso de uso anterior.

Normalmente, para corrigir modelos de sistemas que possuem Casos de Uso de Negócios, dividem-se os Casos de Uso de Negócios em alguns Casos de Uso de Sistema. Para cada Caso de Uso de Negócios, é necessário identificar fluxos que representem valores de negócio. Cada um desses fluxos será extraído para novos Casos de Uso de Sistema.

Um comportamento que facilita a identificação desses novos casos de uso, é avaliar se existe um ator que está enviando alguma informação que não tenha sido requisitada pelo sistema. Isso é um forte indício, apesar de não ser regra geral, de que um novo caso de uso deve ser criado. Outro comportamento a ser identificado é avaliar atrasos entre as ações do fluxo do Caso de Uso de Negócios. O padrão *Seqüência de Casos de Uso*, que está detalhado nesse trabalho, pode auxiliar nessa atividade de transformação dos Casos de Uso de Negócios.

3.1.6 Mistura de níveis de abstração

Misturar em um mesmo modelo, casos de uso que estão em níveis de abstração diferentes é uma falha bastante comum. Um exemplo que pode ser citado é o modelo de uma empresa de telefonia que contém os casos de uso *Realizar Chamada Telefônica* e *Verificar Ruído de Fundo Usando Tronco X*. O primeiro caso de uso modela um fluxo geral de um nível de abstração mais alto, enquanto o segundo realiza a modelagem em um nível de abstração mais baixo, com muito detalhamento.

Modelos com casos de uso em diferentes níveis de abstração são difíceis de ler e entender, assim como realizá-los e implementá-los. Além disso, algumas partes dos serviços do sistema serão enfatizadas justamente por serem mais elaboradas, enquanto outras partes, que podem ser vistas como menos importantes, serão modeladas brevemente.

Gerentes de sistema e usuários provavelmente apreciarão casos de uso em um nível de abstração alto e certamente terão dificuldade em entender casos de uso em níveis de abstração mais baixo. Por outro lado, desenvolvedores irão considerar os casos de uso que apresentam nível de abstração mais alto, insuficientes para começar o trabalho de projeto.

Para detectar a falha de mistura de níveis de abstração, deve-se analisar a descrição dos casos de uso de um modelo, verificando em quais deles ações genéricas e conceitos abstratos são descritas e aqueles onde descrições mais detalhadas e conceitos mais concretos podem ser encontrados. Para facilitar essa detecção, um projetista pode fazer a revisão do modelo, verificando se o nível de detalhamento é suficiente para permitir a realização dos casos de uso para posterior montagem do projeto. O projetista pode então apontar quais casos de uso estão em um nível de abstração muito alto e solicitar um maior detalhamento, assim como apontar os casos de uso em um nível de abstração mais baixo, que já podem ser utilizados para o projeto. Esse mesmo tipo de revisão pode ser feita pelos *Stakeholders* ligados ao negócio e também pelos usuários do sistema, que irão analisar os casos de uso dentro da sua visão de necessidade de uso.

3.1.7 Múltiplos valores de negócio

Na tentativa de diminuir o tamanho de um modelo de casos de uso, desenvolvedores “somam” casos de uso que antes estavam separados no modelo. Cada caso de uso deve possuir somente um objetivo, facilitando para o leitor entender a proposta e o valor do caso de uso. Quando um caso de uso possui muitos objetivos, por consequência, existirão muitos *Stakeholders* interessados no caso de uso, tornando mais complicada a tomada de decisões. Além disso, recursos também poderão ser mal aproveitados; por exemplo, um revisor de casos de uso será obrigado a ler um fluxo que possui objetivos que não o interessam. Para exemplificar, pode-se considerar um caso de uso que tem como objetivos contratar, demitir e mover um funcionário para outro setor da empresa.

Dividir um caso de uso não aumenta o trabalho de desenvolvimento. Certamente existirá um aumento no número de casos de uso do modelo, assim como um aumento na documentação necessária para descrevê-los, mas por outro lado, o entendimento do modelo irá aumentar.

Para resolver esse tipo de falha de modelagem, deve-se basicamente dividir os casos de uso do tipo “Faça qualquer coisa” em outros casos de uso, que tenham cada um deles um único objetivo ou valor de negócio. Como passo inicial, os atores envolvidos no caso de uso deverão ser revisados. Muitas vezes, por consequência da falha de modelagem, existem atores representando vários papéis. O próximo passo, então, é focar em um ator por vez. A parte dos fluxos dos casos de uso originais em que os atores estão associados serão candidatos a constituírem novos casos de uso. Alguns desses casos de uso candidatos poderão ser combinados em apenas um, aplicando o padrão CRUD, conforme apresentado mais à frente, nesse trabalho. Uma vez que todas as descrições relevantes foram movidas para os novos casos de uso, os casos de uso originais deverão ser removidos do modelo.

3.1.8 Níveis de Controle de Acesso como Atores

Atores de um sistema estão muito próximos dos papéis associados ao negócio e todos esses papéis possuem seus próprios níveis de acesso. Em consequência disso, esses atores se relacionam aos serviços do sistema. Dessa forma, torna-se tentador definir os níveis de acesso, relacionando-os diretamente como atores.

Entretanto, a definição de um ator deve ser lembrada: “Um ator é um papel representado por um usuário externo do ponto de vista do sistema”. Esse papel não pode ser escolhido por um usuário externo. Isso implica que, assim que uma pessoa ou entidade externa ao sistema inicia um determinado caso de uso, essa pessoa ou entidade irá, automaticamente, representar o papel modelado pelo ator iniciador do caso de uso.

Capturar requisitos de acesso em um modelo de casos de uso, somente conectando níveis de acesso para atores, significa que qualquer um que esteja usando o caso de uso associado, terá o nível de acesso do ator sob o ponto de vista do sistema. Esse cenário não representa corretamente os requisitos de controle de acesso da maioria dos sistemas. A Fig. 3.4 apresenta o cenário descrito.

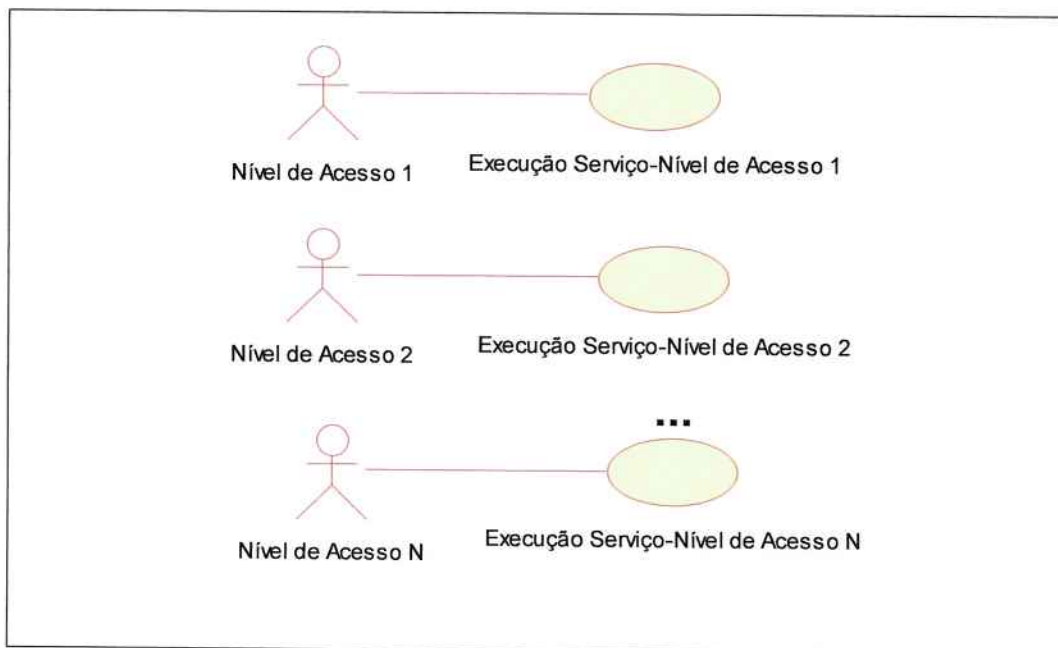


Fig. 3.4 – Atores modelando níveis de acesso

Para corrigir esse tipo de problema em um modelo, pode-se aplicar o padrão *Direitos de Acesso*, conforme detalhado nesse trabalho.

3.2 Padrões (*Patterns*)

Padrões para casos de uso estabelecem soluções para problemas de modelagens (OVERGAARD e PALMKVIST, 2004). Os padrões pesquisados e descritos nesse trabalho foram divididos em duas categorias:

- *Templates*: padrões que podem ser aplicados para domínios genéricos dentro de um determinado contexto;
- Domínio Específico: padrões que podem ser aplicados para domínios específicos dentro de um determinado contexto.

Dentro de cada uma das categorias, estão listados grupos de padrões. Cada grupo de padrões engloba um ou mais padrões relacionados a um contexto. Por exemplo, o grupo *Múltiplos Atores* engloba os padrões *Papéis Diferentes* e *Papel Comum*. Ambos os padrões referem-se ao contexto relacionado à utilização de vários atores para um mesmo caso de uso.

Os padrões com as respectivas categorias e grupos estão divididos conforme a Fig. 3.5 e Fig. 3.6.

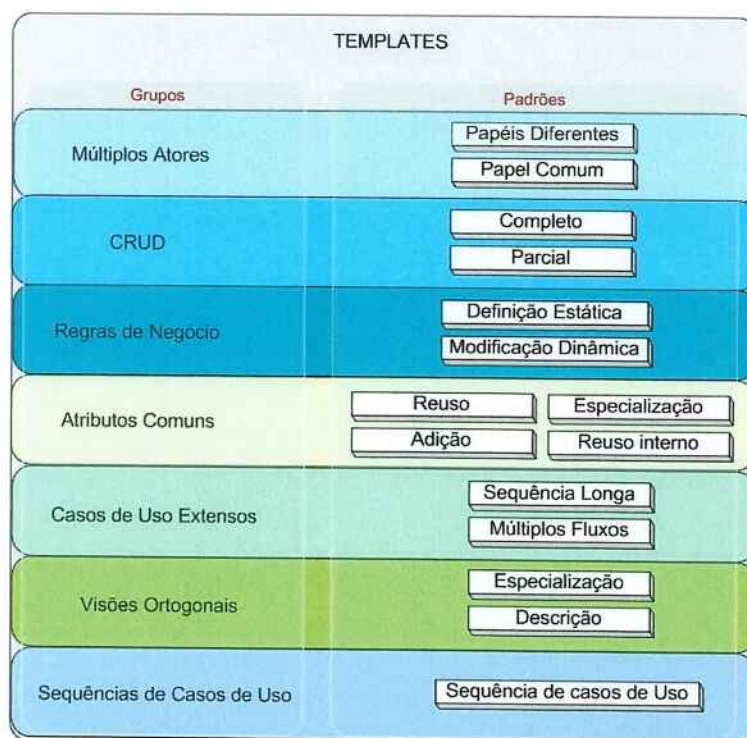


Fig. 3.5 – Padrões de casos de uso para a categoria *Templates*

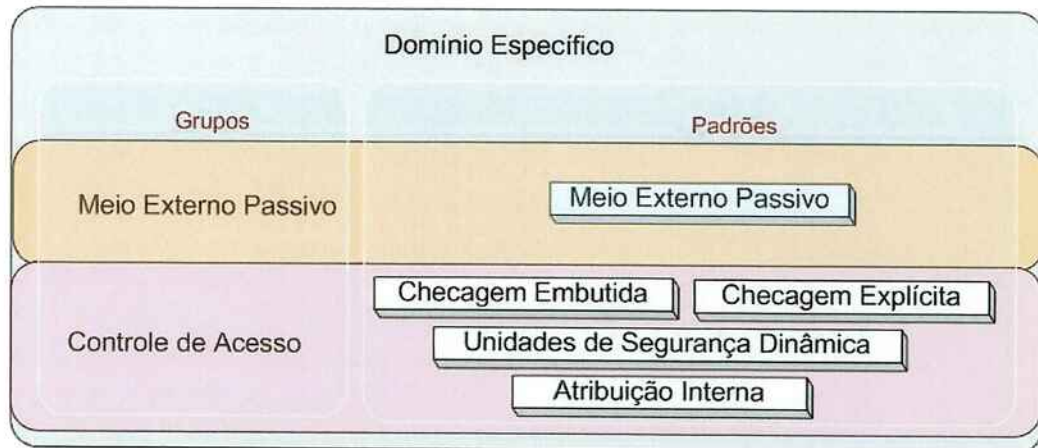


Fig. 3.6 – Padrões de casos de uso para a categoria Domínio Específico

A descrição dos padrões segue um formato com as seguintes seções:

Discussão: descreve a análise envolvida para definição do padrão. Essa seção pode ocorrer para um grupo de padrões ou especificamente para um padrão. No caso de um grupo de padrões, a discussão é válida para todos os padrões.

Parte do caso de uso afetado: a aplicação do padrão pode afetar duas partes em um caso de uso:

- **Modelo:** quando a estrutura do Diagrama de Casos de Uso é afetada.
- **Descrição:** quando a estrutura descritiva de um ou mais casos de uso é afetada.

Modelo: diagrama UML, representando o modelo relacionado ao padrão. Essa seção só está disponível para os padrões que afetam o Modelo de Casos de Uso (Seção – Parte do caso de uso afetado = “Modelo”)

Aplicabilidade: descreve a situação em que o padrão pode ser aplicado.

Exemplo: mostra um exemplo para melhorar o entendimento do padrão discutido. O Exemplo pode ser aplicado para o grupo de padrões ou especificamente para um padrão.

3.3 Catálogo de padrões – Templates

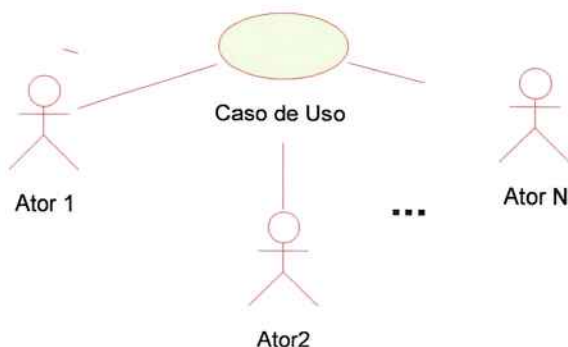
3.3.1 Grupo – Múltiplos Atores

Os padrões nesse grupo, como o próprio nome o diz, envolvem casos de uso onde aparecem mais de um ator. O grupo é formado pelos padrões:

- Papéis diferentes
- Papel comum

Tabela 3.1 – Padrão *Papéis Diferentes*

<i>Padrão: Papéis Diferentes</i>	<i>Categoria: Templates</i>
Parte do caso de uso afetada: Modelo	
<p>Discussão: Quando se cria um modelo, é comum identificar dois ou mais atores que se relacionam com o mesmo caso de uso. Normalmente, isso significa que existem múltiplas entidades externas interessadas em utilizar o sistema conforme modelado pelo caso de uso. Entretanto, nessas situações é importante verificar se os atores representam papéis idênticos no caso de uso ou se os papéis diferem. Essa verificação é importante, não somente para entender os detalhes do caso de uso, mas também para definir as interfaces do sistema. É necessário estabelecer, do ponto de vista do caso de uso, quantas instâncias externas irão interagir com cada instância do caso de uso.</p> <p>Se duas ou mais instâncias externas estão interagindo de maneira diferente com a instância do caso de uso, pode-se definir um ator para cada uma dessas instâncias, relacionando-os com o caso de uso.</p> <p>Suponha uma chamada telefônica, atendida por um caso de uso denominado <i>Chamada Local</i>. O ator Assinante Chamador inicia o caso de uso, disca o número desejado e aguarda que sua chamada seja atendida. O ator Assinante Chamado, por sua vez, simplesmente atende a chamada. Fica evidente que o ator Assinante Chamador e o ator Assinante Chamado agem de forma diferente do ponto de vista do caso de uso e, dessa forma, está correta a utilização de dois atores para o mesmo caso de uso.</p>	

Modelo:

Aplicabilidade: É usado quando dois atores representam papéis diferentes em um caso de uso, ou seja, eles interagem de maneira diferente com o caso de uso.

Exemplo: Caso de uso *Chamada Local*

Descrição resumida: O Assinante Chamador faz uma chamada para o Assinante Chamado em uma mesma rede telefônica.

Atores: Cliente Chamador e Cliente Chamado.

Fluxo Básico:

1. O caso de uso inicia quando o Assinante Chamador tira o telefone do gancho.
2. Registra-se o Assinante Chamador com o estado "Ocupado" e manda-se um tom de discagem para o Assinante Chamador.
3. O Assinante Chamador discar o número telefônico e o sistema analisa, a cada dígito discado, se o número recebido tem a quantidade de dígitos suficientes para determinar a direção da ligação. Enquanto isso não ocorrer, o sistema aguarda novos dígitos.
4. Identifica-se o Assinante Chamado e requisita-se à rede telefônica uma conexão entre o Assinante Chamador e o Assinante Chamado.
5. Um tom de "chamando" é enviado para o Assinante Chamador e um sinal de "chamada" é enviado para o Assinante Chamado. O telefone do Assinante Chamado "toca".
6. O Assinante Chamado atende a ligação. O sistema interrompe o tom de "chamando" e o sinal de "chamada". O Assinante Chamado recebe o estado "Ocupado".
7. Quando o Assinante Chamador e o Assinante Chamado terminam a ligação, ou seja, quando ambos colocam o telefone no gancho, o sistema requisita à rede telefônica a desconexão da ligação.
8. O caso de uso registra o Assinante Chamador e o Assinante Chamado com estado de "Aguardando ligação".
9. O caso de uso termina.

Fluxos Alternativos:

- A qualquer momento, antes do Assinante Chamado atender uma ligação, a ligação pode ser descontinuada pelo Assinante Chamador. Caso a rede telefônica já tenha determinado uma conexão para essa ligação, o sistema solicita a desconexão e registra o Assinante Chamador e o Assinante Chamador com o estado "Aguardando ligação".
- Quando uma ligação está em andamento, qualquer um dos participantes pode interromper a ligação, ou seja, um ou outro podem colocar o telefone no gancho. Nesse caso, sua comunicação é interrompida, mas a conexão da rede telefônica é mantida, enquanto o outro estiver com o telefone

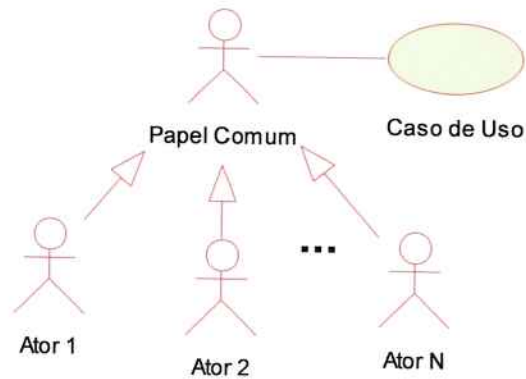
fora do gancho. A parte que interrompeu a ligação (Assinante Chamador ou Assinante Chamado) pode, posteriormente, tirar o telefone do gancho. Nesse caso a comunicação é restabelecida e a ligação é continuada.

- No passo 2: O Assinante Chamador não tem autorização para fazer ligações, naquele momento. A ligação não é efetuada.
- No passo 3: O Assinante Chamador não completa o total de dígitos dentro do espaço de tempo estabelecido pelo sistema. A conexão é interrompida por *time-out*.
- No passo 5: O Assinante Chamado não atende a ligação após um número especificado de toques (15 toques, por exemplo), sendo a chamada interrompida. O Assinante Chamante recebe o sinal de "Ocupado", marcando a desconexão da linha telefônica.
- No passo 5: O Assinante Chamado está no estado "Ocupado". O Assinante Chamante recebe o sinal de "Ocupado" e a conexão é interrompida. O Assinante Chamante deve colocar o telefone no gancho para voltar ao estado "Aguardando ligação".

Casos de Uso relacionados: nenhum.

Tabela 3.2 – Padrão *Papel Comum*

<i>Padrão: Papel Comum</i>	<i>Categoria: Templates</i>
Parte do caso de uso afetada: Modelo	
<p>Discussão: Em casos de uso onde existem múltiplos atores associados com o mesmo caso de uso, é comum ter-se a impressão de que múltiplas entidades externas interagem com cada instância do caso de uso. Essa impressão muitas vezes é incorreta e pode gerar modelos errados. Em tais casos, não existem múltiplos atores, mas sim, múltiplos papéis.</p> <p>Suponha-se a venda de passagens aéreas e que, um funcionário da companhia aérea e um agente de viagens externo estão autorizados a vender passagens para clientes da companhia. Nesse caso, ambos podem solicitar uma passagem, mas do ponto de vista do caso de uso, existe somente um tipo de entidade externa interagindo com cada instância do caso de uso. O funcionário e o agente de viagens representam o mesmo papel para o caso de uso, pois agem exatamente da mesma forma. Assim, somente um ator deve estar associado. Supondo que os atores Funcionário e Agente Externo já tenham sido identificados no modelo (já são utilizados em outros casos de uso), torna-se necessário introduzir um novo ator abstrato, chamado nesse exemplo de Vendedor de Passagens. Esse novo ator deve ser associado ao caso de uso <i>Venda de Passagens</i> e os atores Funcionário e Agente Externo devem relacionar-se a ele através de um relacionamento de herança (Generalização e Especialização). Se Funcionário e Agente Externo não são atores necessários para outros casos de uso, apenas o ator Vendedor de Passagens pode ser utilizado.</p>	

Modelo:

Aplicabilidade: É aplicável quando, do ponto de vista do caso de uso, existe somente uma entidade externa interagindo com cada instância do caso de uso.

Exemplo: Caso de Uso *Venda de Passagens Aéreas*

Descrição Resumida: O Vendedor de Passagens registra, no sistema, uma nova passagem para os vôos previamente agendados pela companhia aérea.

Ator: Vendedor de Passagem

Fluxo Básico:

1. O caso de uso inicia quando o Vendedor de Passagens precisa registrar uma nova passagem.
2. O sistema solicita ao Vendedor de Passagens o nome e endereço do cliente.
3. O Vendedor de Passagens fornece as informações solicitadas.
4. O sistema informe ao Vendedor de Passagens o *status* do cliente (especial ou normal) e quantas passagens ele comprou nos últimos doze meses.
5. O Vendedor de Passagens indica quais vôos devem ser incluídos na passagem – com suas respectivas origens e destinos – e para isso, fornece o número do vôo, a origem, o destino, a data e o horário de partida ou se um vôo em específico deve ser pesquisado.
6. Para cada vôo incluído: o sistema envia uma solicitação de reserva para a companhia aérea. A companhia aérea retorna o número de confirmação da reserva, que é incluído na reserva da passagem.
7. O caso de uso apresenta para o Vendedor de Passagens todas as informações relacionadas à reserva e solicita sua confirmação.
8. O Vendedor de Passagens confirma a reserva.
9. O sistema registra uma nova passagem e armazena o nome e endereço do cliente junto às informações do vôo, assim como o nome do Vendedor de Passagens.
10. O número da reserva é mostrado para o Vendedor de Passagens.
11. O Vendedor de Passagens fornece o número da reserva ao cliente.

Fluxos Alternativos:

- No passo 4: O cliente não está cadastrado no sistema. Nesse caso, ele tem que efetuar um cadastro junto à empresa aérea. O caso de uso de *Cadastramento de cliente* deve ser executado.
- No passo 5: O Vendedor de Passagens não conhece os dados do vôo a ser inserido para a reserva da passagem. Para determinar esses dados, ele executa o caso de uso *Procurar Vôo* que deve ser incluído. Caso o resultado consista de vários vôos, o sistema solicita que o Vendedor de Passagens

selecione um deles. O Vendedor de passagens escolhe um voo e o adiciona à reserva.

- No passo 6: A companhia aérea retorna uma falha de reserva, o Vendedor de Passagens é notificado e o voo não é incluído na reserva. Um novo voo deve ser determinado e para isso, deve-se voltar ao passo 5.
- No passo 6: o Vendedor de Passagens fica sem conexão com a companhia aérea. O Vendedor de Passagens é notificado e o voo não é incluído na passagem.
- No passo 7: o Vendedor de Passagens escolhe não confirmar a reserva, e o sistema solicita a confirmação para o seu cancelamento. Caso confirmada, a reserva é removida do sistema e o caso de uso é encerrado. Para a remoção da reserva, é utilizado o caso de uso *Desfazer Reserva*. Senão, vai-se ao passo 8.
- A qualquer instante, antes do Passo 8: a reserva pode ser cancelada pelo Vendedor de Passagens e todas as informações removidas do sistema. Caso o cliente tenha já realizado o seu cadastro junto ao sistema, este não é removido junto com as demais informações.

Casos de Uso relacionados:

- *Cadastramento de Cliente* (incluído), *Procurar Voo* (incluído) e *Desfazer Reserva* (de Extensão).

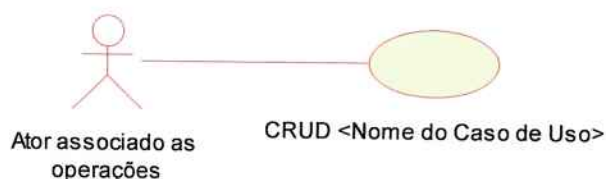
3.3.2 Grupo – CRUD

Operações para criação (*Create*), leitura (*Read*), atualização (*Update*) e remoção (*Remove*) de informações (CRUD – Iniciais em inglês) são extremamente comuns em sistemas de informação. Normalmente são operações simples que ao serem processadas exigem uma simples verificação de sintaxe e tipo ou em outros casos algum tipo de cálculo trivial ou checagem de uma regra de negócio. A descrição do fluxo possui apenas algumas sentenças e geralmente não existe mais do que dois ou três caminhos alternativos no fluxo.

Muitos profissionais questionam se realmente precisam criar e incluir casos de uso para operações do tipo CRUD na modelagem do sistema. A criação desses casos de uso é necessária, pois as operações CRUD alteram o estado do sistema e produzem um resultado observável para os atores que interagem com esses casos de uso. A não inclusão desses casos de uso gera um modelo incompleto e provavelmente o sistema final não atenderá alguns requisitos.

Tabela 3.3 – Padrão *CRUD Completa*

Padrão: <i>CRUD Completa</i>	Categoria: <i>Templates</i>
Parte do caso de uso afetada: Modelo	
Discussão: A utilização do padrão CRUD: Completo faz com que todas as quatro operações (<i>Create</i> , <i>Read</i> , <i>Update</i> e <i>Delete</i>) para um determinado tipo de informação sejam agrupadas em um único caso de uso chamado CRUD <Nome do caso de uso>.	
Um Caso de uso CRUD também pode incluir outros fluxos básicos além dos quatro fluxos; por exemplo, fazer pesquisa de um item ou fazer algum cálculo simples, baseado em algum item.	

Modelo:**Aplicabilidade:**

Esse padrão pode ser utilizado quando todos os fluxos relacionados as operações do tipo CRUD contribuem para o mesmo valor de negócio. Além disso, esses fluxos devem ser curtos e simples.

Exemplo detalhado:

Como exemplo, suponha-se um sistema para gerenciamento de atividades em geral. Ele deve permitir o registro de novas atividades que devem ser realizadas em um dado momento no futuro, modificar atividades ainda não realizadas, cancelar uma atividade e apresentar tarefas que falharam durante a execução. Nota-se nessa descrição, quatro operações básicas que podem ser descritas em quatro fluxos básicos. Tratamento de erros e exceções serão tratados como fluxos alternativos do caso de uso.

Exemplo: CRUD Atividades

Descrição Resumida: O caso de uso registra, modifica ou cancela uma atividade de acordo com as informações fornecidas pelo Definidor de Tarefas. Além disso, permite a visualização de atividades que falharam durante sua execução.

Ator: Definidor de Atividades

Fluxo Básico:

O caso de uso tem quatro diferentes fluxos básicos:

- Registrar uma nova atividade
- Modificar uma atividade que já existe
- Cancelar uma atividade
- Ver atividades que falharam

Registrar uma nova atividade

1. O caso de uso inicia quando o Definidor de Atividades escolhe registrar uma nova atividade.
2. O caso de uso apresenta uma lista de possíveis tipos de atividades para o Definidor de Atividades e pergunta o tipo de atividade que será registrada, o nome a ser atribuído e quando será realizada.
3. O Definidor de Atividades entra com as informações requisitadas. O caso de uso checa se o dia informado para realização da atividade está no futuro e se o nome da tarefa é único.
4. O caso de uso registra a nova atividade no sistema e marca a tarefa com o estado "Habilitada"
5. O caso de uso termina.

Modificar uma atividade que já existe

1. O caso de uso começa quando o Definidor de Tarefas escolhe modificar uma atividade que já existe (uma atividade registrada). O caso de uso busca o nome de todas as atividades que não estão marcadas como “Ativas” e apresenta para o Definidor de Atividades.
2. O Definidor de Atividades seleciona uma das atividades. O caso de uso busca informações sobre a tarefa selecionada e apresenta para o definidor de atividades.
3. O Definidor de Atividades modifica qualquer informação apresentada com exceção do nome da atividade.
4. O Definidor de Atividade confirma as modificações. O caso de uso checa se a data informada está no futuro, caso afirmativo, armazena a informação modificada.
5. O caso de uso termina.

Cancelar uma atividade

1. O caso de uso começa quando o Definidor de Atividades escolhe cancelar uma atividade. O caso de uso busca todas as atividades que não estão marcadas como ativas e apresenta para o Definidor de Atividades.
2. O Definidor de Atividades seleciona uma das atividades. O caso de uso busca informações sobre a tarefa selecionada e apresenta para o Definidor de Atividades.
3. Se o Definidor de Atividades confirma o cancelamento, o caso de uso remove a atividade, caso contrário nenhuma alteração é realizada.
4. O caso de uso termina.

Ver atividades que falharam

1. O caso de uso começa quando o definidor de atividades escolhe ver a lista de atividades que falharam.
2. O caso de uso coleta todas as atividades com o estado “Falha” e apresenta seus nomes para o definidor de atividades.
3. O caso de uso termina.

Fluxos alternativos

- a qualquer instante – Abortar operação: O Definidor de Atividades pode escolher a qualquer momento abortar a operação que está realizando no momento, nesse caso, qualquer modificação dentro da operação deve ser descartada e o caso de uso é encerrado.
- Nos fluxos básicos “Registrar uma nova atividade” ou “Modificar uma atividade que já existe”- Se o nome da atividade não é único ou a data da atividade não está no futuro, o definidor de atividade é notificado que a informação está incorreta e é requisitado para informar novamente os dados.

Casos de Uso relacionados: nenhum.

Tabela 3.4 – Padrão *CRUD Parcial*

Padrão: <i>CRUD Parcial</i>	Categoria: <i>Templates</i>
Parte do caso de uso afetada: Modelo	
<p>Discussão: Caso algumas das quatro operações CRUD sejam mais simples e outras sejam mais complexas, é recomendado que as operações mais simples sejam agrupadas em um caso de uso e as mais complexas sejam modeladas em casos de uso separados. As situações mais complexas, normalmente aparecem quando existe mais de um fluxo básico para uma determinada operação. Um caso de uso CRUD típico possui quatro fluxos básicos (um para cada operação) e alguns fluxos alternativos.</p> <p>O mais importante é não agrupar operações complexas dentro de um único caso de uso. Essas operações deverão ser tratadas isoladamente, pois provavelmente serão projetadas e desenvolvidas separadamente.</p> <p>Como regra geral, quando não se tem certeza em agrupar ou manter isoladas as operações CRUD, deve-se mantê-las separadas. Essa decisão não afetará a funcionalidade do sistema, somente afetará a estrutura do modelo e em alguns casos a manutenção do sistema.</p>	
Modelo:	
<p>O diagrama mostra um ator (stick figure) rotulado "Ator associado as operações" conectado por duas linhas a dois casos de uso. O caso de uso superior é um oval verde rotulado "CRUD <Nome do Caso de Uso>". O caso de uso inferior é um oval verde rotulado "Operação X".</p>	
Aplicabilidade:	
Esse padrão deve ser empregado, preferencialmente, quando um dos casos de uso alternativos é mais significativo, mais longo ou mais complexo do que as outras alternativas.	
Exemplo: Não apresentado.	

3.3.3 Grupo – Regras de Negócio

Muitas organizações como bancos, hospitais e seguradoras possuem normas, políticas e melhores práticas que são utilizadas em seus negócios. Esses tipos de regras, que nesse trabalho são referenciadas simplesmente como Regras de negócio, podem ser definidos pela própria organização ou externamente pelo Governo, por clientes ou por padronizações de mercado. Qualquer sistema desenvolvido ou utilizado por essas organizações deve seguir e respeitar essas regras.

Suponha uma empresa que trabalhe como uma central de pedidos. Essa empresa possui várias regras de negócio associadas aos pedidos. Algumas delas poderiam ser:

1. Um pedido é uma aceitação oficial entre um cliente e a empresa, definindo que o cliente está comprando uma coleção de itens da empresa para serem entregues e pagos em uma determinada data;
2. Um pedido pode conter múltiplos itens;
3. Um pedido deve conter pelo menos um item;
4. O valor do pedido é o somatório dos valores de todos os itens incluídos, mais os custos de entrega e respectivos tributos;
5. Um pedido rápido é um pedido feito antes das seis da tarde e entregue até as quatro da tarde do dia posterior ao pedido;
6. Uma fatura é gerada e enviada depois que o pedido foi entregue.

Essas regras podem ser escritas e organizadas de várias maneiras. O caminho mais simples é enumerá-las em um documento e fornecer a cada uma delas um nome único ou uma identificação. Outro modo é definir as regras de negócios em um modelo de classes. Nesse modelo, a regra 1 citada no exemplo, poderia ser expressa como uma classe, enquanto a regra 2 poderia ser uma associação nesse modelo. O restante das regras poderia ser definido como funcionalidades dos elementos do modelo.

As regras de negócio expõem a essência do negócio, independente de como são desenvolvidas. A separação da especificação das regras de negócio facilita alterações no modelo. As regras de negócio são expostas, explicitamente, ao invés de ficarem misturadas no meio de ações que não se relacionam.

O primeiro padrão – *Definição Estática* – assume que as regras de negócio estão misturadas com outras camadas do sistema e o segundo padrão – *Modificação Dinâmica*, fornece um caso de uso para atualização das regras, enquanto o sistema está em uso. A escolha de um dos padrões citados vai depender da possibilidade de se modificar regras dinamicamente. O caso de uso que modela a criação e manutenção das regras de negócios é descrito usando técnicas comuns de caso de uso, ou seja, não existe nenhum formato especial para esse caso de uso.

Para beneficiar-se completamente das regras de negócios, algumas empresas não só descrevem as regras separadamente, mas também as implementam separadamente do resto do código que utiliza essas regras. A implementação das regras de negócios pode ser feita em componentes separados, repositórios, produtos de terceiros, etc.

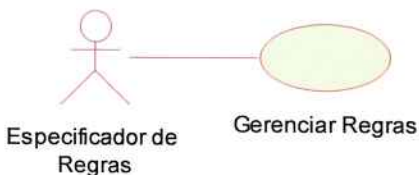
Dessa forma, uma regra pode ser modificada sem a necessidade de se alterar o resto da aplicação.

Os padrões para regras de negócio também podem ser aplicadas para outros tipos de informações recorrentes ou que são repetidamente referenciadas nas descrições dos casos de uso. Exemplos incluem lista de parâmetros e atributos, definições, informações para serem impressas em cabeçalhos, etc.

Tabela 3.5 – Padrão *Definição Estática*

Padrão: <i>Definição Estática</i>	Categoria: <i>Templates</i>
Parte do caso de uso afetada: Descrição	
<p>Discussão: Quando se cria um modelo de casos de uso de um sistema que possui restrições de regras de negócios, os casos de uso são identificados da maneira usual. Entretanto, a descrição desses casos de uso sofre uma significativa alteração. Primeiro, ao invés de descrever as diferentes decisões de negócios na descrição do fluxo, referências são feitas dessa descrição para a coleção de regras de negócios. Segundo, as regras de negócio não são descritas no mesmo documento dos casos de uso, elas são descritas em um documento separado ou em algum tipo de repositório. Dessa forma, as regras de negócio são descritas apenas uma vez podendo ser aplicada para vários casos de uso.</p> <p>Suponha o seguinte fragmento do caso de uso <i>Criar pedido</i>, relacionado ao endereço de entrega de um pedido:</p> <p>“Ao salvar um pedido, o endereço de entrega é verificado de acordo com a regra BR10. Se a regra foi cumprida, as informações sobre o pedido são armazenadas no sistema; caso contrário, o Criador de Pedidos é requisitado para entrar com um novo endereço de entrega.”</p> <p>A regra de negócio é definida como:</p> <p>BR10. O Endereço de entrega de um pedido deve conter um nome, rua e complementos, cidade, CEP e estado do receptor.</p> <p>Com essa abordagem, as regras de negócio podem ser lidas e atualizadas separadamente dos casos de uso. Além disso, os casos de uso podem ser revisados e aprovados independentes da aprovação das regras. Do ponto de vista de gerenciamento do projeto, essa abordagem é um grande benefício, pois a descrição das regras de negócio e dos casos de uso tem ciclos de desenvolvimento diferentes e a aprovação formal das regras envolve pessoas de negócio que normalmente não estão ligadas diretamente ao projeto (por exemplo, advogados ou especialistas em algum tipo de padronização).</p>	
Aplicabilidade:	
Esse padrão é apropriado quando não existe necessidade de alterações dinâmicas nas regras de negócios, enquanto o sistema está em uso.	
Exemplo: Não apresentado.	

Tabela 3.6 – Padrão *Modificação Dinâmica*

Padrão: <i>Modificação Dinâmica</i>	Categoria: <i>Templates</i>
Parte do caso de uso afetada: Modelo e Descrição	
Discussão: O modelo desse padrão apresenta um caso de uso, chamado <i>Gerenciar Regras</i> que cria, atualiza e remove regras de negócio (pode-se aplicar o padrão CRUD conforme descrito nesse trabalho).	
Modelo:	
 <pre> graph LR Actor[Especificador de Regras] --- UseCase((Gerenciar Regras)) </pre>	
Aplicabilidade:	
Esse padrão é útil quando regras de negócios podem ser modificados dinamicamente, ou seja, podem ser alteradas enquanto o sistema estiver em uso.	
Exemplo: Não apresentado.	

3.3.4 Grupo – Atributos Comuns

Em muitos sistemas existem subsequências de ações que são as mesmas em vários casos de uso. Se isso é apenas uma coincidência e essas subsequências não possuem relação uma com as outras e, além disso, podem ser modificadas independentemente, nenhuma análise quanto a aplicação de padrões precisa ser feita. Entretanto, se é requerido que essas subsequências sejam as mesmas em um dado modelo, é necessário que seja feita uma análise mais detalhada.

A subsequência comum de ações pode ser modelada, incluindo seus fluxos alternativos, em um caso de uso separado. Quando uma sequência precisa ser modificada, apenas um caso de uso será atualizado. O caso de uso deve conter relacionamentos de um mesmo tipo entre a modelagem da subsequência de ações e os casos de uso em que a subsequência aparecem. O tipo de relacionamento (Inclusão, Extensão e Generalização) depende da situação, conforme descrito em cada padrão dessa seção.

Tabela 3.7 – Padrão *Reuso*

Padrão: <i>Reuso</i>	Categoria: <i>Templates</i>
Parte do caso de uso afetada: Modelo e Descrição	
<p>Discussão: Se uma subseqüência comum de ações deve ser independente do contexto em que deverá ser utilizada, um relacionamento de inclusão poderá ser utilizado. Nesse caso, os casos de uso base deverão ter relacionamentos de inclusão para o caso de uso que modela a subseqüência de ações. A descrição do fluxo do Caso de Uso Base deverá incorporar uma referência para o caso de uso de inclusão onde a subseqüência será realizada, mas não deverão existir referências para outras direções, pois a subseqüência comum é independente de seu contexto. O reuso de uma mesma subseqüência de ações em múltiplos casos de uso é um exemplo típico de quando usar um relacionamento do tipo inclusão.</p> <p>Em um sistema de biblioteca, o usuário de livros deve sempre apresentar seu cartão da biblioteca e sua senha quando solicita um novo empréstimo, quando verifica empréstimos pendentes ou quando deseja estender um empréstimo existente.</p> <p>A subseqüência que verifica se o cartão é válido assim como a senha informada deverá ser modelada em um caso de uso separado para ser utilizado em todos os diferentes casos de uso que manipulam empréstimos. Quando uma nova transação de empréstimo precisa ser incluída para o sistema da biblioteca, o novo caso de uso terá um relacionamento de inclusão para o caso de uso que já existe chamado <i>Verifica Cartão e Senha</i>. Dessa forma, a seqüência de ações será reutilizada pelo novo caso de uso.</p> <p>Toda a subseqüência do caso de uso comum deve ser executada pelo Caso de Uso Base. Caso exista necessidade de que partes diferentes da subseqüência sejam executadas de acordo com o Caso de Uso Base, novos relacionamentos do tipo inclusão deverão ser criados para cada parte da subseqüência.</p>	
Modelo:	
Aplicabilidade:	
<p>Nesse padrão, a subseqüência representa uma parte e deve ser incluída nos outros casos de uso para formar um todo. Nenhuma referência deve ser feita da subseqüência para onde ela será usada, pois casos de uso do tipo inclusão devem ser independentes dos Casos de Uso Base.</p>	
Exemplo: Não apresentado.	

Tabela 3.8 – Padrão *Adição*

Padrão: <i>Adição</i>	Categoria: <i>Templates</i>
Parte do caso de uso afetada: Modelo e Descrição	
<p>Discussão: Quando uma nova subseqüência de ações é adicionada para vários casos de uso que já existem ou quando a mesma subseqüência pode ser extraída dos casos de usos que já existem sem torná-los incompletos, um relacionamento de extensão poderá ser utilizado. Nesse contexto, casos de uso que já existem serão independentes da subseqüência comum de ações e dessa forma não conterão qualquer referência para ela. Sendo assim, a subseqüência poderá ser adicionada ou removida sem afetar as definições de qualquer caso de uso que já exista.</p> <p>Supondo ainda o exemplo da biblioteca, cada vez que um livro é devolvido para a biblioteca (caso de uso <i>Retorna Livro</i>), o sistema verifica se alguém fez um pré-registro para o próximo empréstimo desse livro. Se isso acontecer, uma notificação é enviada para a pessoa e o livro torna-se indisponível para qualquer outro empréstimo pelos próximos três dias. A mesma verificação é realizada quando uma nova cópia de livro é comprada pela biblioteca e registrada no sistema (caso de uso <i>Cadastra Nova Cópia</i>). Como os casos de uso <i>Retorna Livro</i> e <i>Cadastra Nova Cópia</i> são completos por si só, pode-se modelar a verificação da reserva em um caso de uso separado chamado <i>Fila de Reserva</i> que deverá possuir um relacionamento de extensão para os outros dois casos de uso.</p> <p>Nem todo o comportamento modelado pelo caso de uso do tipo extensão precisa ser utilizado pelo Caso de Uso Base. Partes diferentes desse caso de uso podem ser utilizadas pelos casos de uso base através do recurso de Pontos de Extensão. Entretanto, caso as partes possuam uma ordem dentro do caso de uso de extensão, elas devem ser inseridas respeitando essa ordem dentro do Caso de Uso Base.</p>	
Modelo:	
Aplicabilidade:	
<p>Esse padrão é preferível quando os outros casos de uso já são completos mais podem utilizar extensões, ou seja, os casos de uso que não sejam aqueles com a subseqüência comum modelam um uso completo do sistema e podem estender seu uso com o caso de uso que possui a subseqüência de ações.</p>	
Exemplo: Não apresentado.	

Tabela 3.9 – Padrão *Especialização*

Padrão: <i>Especialização</i>	Categoria: <i>Templates</i>
Parte do caso de uso afetada: Modelo e Descrição	
<p>Discussão:</p> <p>Alguns sistemas possuem casos de uso que realizam tarefas similares. Um exemplo é um sistema de produção automatizada, onde diferentes tipos de diretivas de produção são executados de forma similar, mas não são idênticas. Como todos os casos de uso modelam a execução de tarefas do mesmo tipo, esses atributos comuns deverão ser capturados pelo modelo. Isso é feito introduzindo um novo caso de uso que irá modelar como geralmente uma tarefa será realizada incluindo todas as ações comuns para todos os tipos de tarefas. Execuções específicas das tarefas são capturadas em casos de uso separados com relacionamento de generalização para o novo caso de uso. Os casos de uso filhos podem especializar ações definidas no Caso de Uso Pai assim como adicionar novas ações.</p>	
<p>Modelo:</p> <pre> graph BT CU1([Caso de Uso 1]) --> TUC([Tipo de Uso Comum]) CU2([Caso de Uso 2]) --> TUC CU3([Caso de Uso N]) --> TUC </pre>	
<p>Aplicabilidade:</p> <p>É aplicável quando usos do sistema modelados pelos casos de uso são de um mesmo tipo e esse tipo deve estar visível no sistema.</p>	
<p>Exemplo detalhado:</p> <p>Nesse exemplo, o caso de uso <i>Realizar Tarefa</i> modela como uma tarefa é realizada de maneira geral, detalhando as ações de como ela é selecionada, registro da ocorrência de sua execução (como a tarefa é executada fica a cargo do caso de uso especializado) e o que acontece quando a tarefa é executada. Os dois outros casos de uso são especializações de <i>Realizar Tarefa</i>. <i>Gerar Lista de Seleção</i> modela como as informações dos itens selecionados do depósito são compiladas. O caso de uso <i>Gerar Base de Faturamento</i> modela como as informações sobre todos os pedidos não faturados são compiladas e enviadas para o ator Sistema Financeiro.</p>	
<p>Exemplo: Caso de uso <i>Realizar Tarefa</i></p> <p>Descrição Resumida: Quando iniciado pelo Gerente de Tarefas, o caso de uso traz uma tarefa pré-cadastrada e baseada nessa informação, realiza sua execução. O caso de uso é abstrato.</p>	

Ator: Gerente de Tarefas

Fluxo Básico:

Início do Fluxo

1. O caso de uso inicia quando o Gerente de Tarefas requisita seu uso.
2. O caso de uso coleta todas as tarefas que não foram realizadas até o momento e que não estão cadastradas como “Ativa”, “Falha” ou “Atendimento Futuro”.
3. A tarefa com prioridade mais alta é selecionada e marcada como *Ativa*.

Execução da Tarefa

1. O caso de uso executa a tarefa

Final de Fluxo

1. Se a tarefa é executada com sucesso, ela é removida.
2. O caso de uso termina.

Fluxos Alternativos

- a qualquer instante – Falha na execução da Tarefa: Se a execução de uma tarefa falha, o caso de uso muda o status da tarefa para *Falha* e armazena-a novamente juntamente com o motivo da falha e o caso de uso é encerrado.

- no fluxo básico “Tarefas inexistentes”: Se não existem tarefas para serem executadas, o caso de uso é encerrado.

Casos de Uso relacionados: Gerar Lista de Seleção e Gerar Base de Faturamento.

Caso de uso: *Gerar Lista de Seleção*

Descrição resumida:

O caso de uso *Gerar Lista de Seleção*, que é uma especialização do caso de uso Realizar Tarefa, gera uma lista de itens para serem selecionados do depósito quando compila um novo envio de pedido. A lista gerada é enviada para o Funcionário do Depósito.

Ator: Gerente de Tarefas

Fluxo Básico:

Início do Fluxo

1. O caso de uso começa como descrito no caso de uso *Realizar Tarefa*.

Executando uma Tarefa

1. O caso de uso coleta todas as ordens que foram atribuídas para a lista de seleção.
2. Para cada pedido, o caso de uso busca em que depósito os itens do pedido podem ser selecionados.
3. O caso de uso calcula o número total de cada item do pedido.
4. O caso de uso imprime a lista para o Funcionário do Depósito com todos os itens, as quantidades de cada item e o local do depósito onde cada item pode ser encontrado.
5. O caso de uso muda os status dos pedidos coletados para “Coletados”.

Final de Fluxo

1. O caso termina conforme descrito no caso de uso *Realizar Tarefa*.

Fluxos Alternativos

No fluxo básico “Executando uma Tarefa” :

- No passo 1 - Sem Pedidos: Se nenhum pedido é atribuído para a lista de seleção, a execução da tarefa falha.
- No passo 2 - Local não encontrado: Se o local de um item não pode ser encontrado, todos os pedidos que contem o item em questão recebem o status “Especial” e são removidos da entrega. O caso de uso continua com os pedidos restantes.

Caso de uso: *Gerar Base de Faturamento***Descrição Resumida:**

Esse caso de uso é uma especialização de *Executar Tarefa* e gera a base para faturamento que é enviada para o ator Sistema Financeiro

Fluxo básico:Início de Fluxo

1. O caso de uso começa como descrito no caso de uso *Realizar Tarefa*.

Execução da Tarefa

1. O caso de uso coleta todos os pedidos que estão marcados como “Entregue”.
2. Para cada um desses pedidos, o caso de uso busca o nome do cliente e endereço assim como o número e valor do pedido incluindo descontos e taxas.
3. A informação coletada é enviada para o Sistema Financeiro.
4. O Sistema Financeiro processa a aceitação das informações de faturamento.
5. O caso de uso muda o status dos pedidos processados para “Faturado”

Final de Fluxo

1. O caso termina conforme descrito no caso de uso *Realizar Tarefa*.

Fluxos Alternativos

No fluxo básico “Executando uma Tarefa”:

- No passo 1 - Sem Pedidos: Se nenhum pedido está marcado como “Entregue”, então nenhuma informação é enviada para o Sistema Financeiro.
- No passo 2 - Informação de faturamento não acessível: Se alguma informação requerida para faturamento não pode ser trazida para um pedido, o status do pedido é alterado para “Faturamento Manual”. O caso de uso continua com os pedidos que restam.
- No passo 4 - Não aceitação de faturamento: Se nenhuma aceitação é recebida dentro de dois minutos ou se uma mensagem relatando que alguma informação foi transmitida de maneira incorreta, os status dos pedidos não são alterados (permanecem os mesmos) e a execução da tarefa falha.

Tabela 3.10 – Padrão *Reuso Interno*

Padrão: <i>Reuso Interno</i>	Categoria: <i>Templates</i>
Parte do caso de uso afetada: Descrição	
Discussão: Em um caso de uso, a mesma subsequência de ações pode aparecer várias vezes, normalmente em diferentes fluxos alternativos, mas algumas vezes no mesmo fluxo. Nesses casos, a subsequência comum não deverá ser extraída para um caso de uso separado, pois isso acaba adicionando mais complexidade para o modelo. A subsequência deve ser descrita em uma seção separada. Ao descrever o fluxo do caso de uso, a seção que foi descrita separadamente será referenciada de todos os locais onde ela deve ser executada.	
Aplicabilidade: Esse padrão é preferível quando uma subsequência com ações comuns aparece em vários locais dentro de apenas um caso de uso do modelo.	
Exemplo: Não apresentado.	

3.3.5 Grupo – Casos de Uso Extensos

Qual o tamanho máximo de um caso de uso? Essa é uma pergunta bastante freqüente e não existe uma resposta do tipo “A descrição de um caso de uso deve variar entre uma e quatro páginas” ou “Para cada estímulo externo deve haver no máximo 10 ações...”. Esses números podem variar bastante de um sistema para outro. Por exemplo, um caso de uso pode simplesmente checar alguma informação de entrada e armazená-la no sistema e, em outras situações, casos de uso podem executar dezenas de ações como resposta a um estímulo de um ator. Dessa forma, uma resposta mais apropriada para essa questão seria: “Um caso de uso deve ser extenso conforme for necessário”.

Uma prática bastante comum é avaliar se um caso de uso possui uma descrição que ocupa mais do que três páginas. Isso pode ser um sinal de que o caso de uso está cobrindo muitas regras de negócio ou está em um nível de abstração muito baixo. Entretanto, essa prática não deve ser considerada como regra, pois existem algumas situações onde não há como fugir de um caso de uso extenso.

Quando o fluxo de um caso de uso é extenso, deve-se aplicar técnicas para estruturar a descrição textual de maneira mais inteligente para facilitar a leitura e o entendimento. Uma técnica é introduzir cabeçalhos da mesma forma quando se escreve um texto casual. Outra técnica é extrair descrições de sub-fluxos para dentro de subseções e referenciá-las a partir da descrição principal.

As duas técnicas apresentadas podem ser combinadas desde que seja mantido o objetivo principal ao descrever casos de uso: A descrição deve ser de simples entendimento e de fácil revisão para todos os envolvidos no projeto. Por essa razão, o

nível de abstração na descrição não deve ser modificado simplesmente para se ter descrições curtas. O mesmo nível de detalhamento deverá ser mantido para todas as descrições em um dado modelo, conforme o antipadrão *Mistura de Níveis de Abstração*.

Tabela 3.11 – Padrão *Seqüência Longa*

Padrão: <i>Seqüência Longa</i>	Categoria: <i>Templates</i>
Parte do caso de uso afetada: Descrição	
Discussão: Esse padrão consiste em extrair descrições de sub-fluxos para dentro de subseções e referenciá-las a partir da descrição principal. Na descrição principal, cada referência às subseções deverá conter um pequeno resumo que indica seu conteúdo. Dessa forma, o leitor terá uma visão geral do fluxo inteiro e poderá estudar as descrições detalhadas quando necessário.	
Aplicabilidade: É aplicável para um caso de uso extenso quando a descrição pode ser estruturada em cabeçalhos ou seções de maneira a facilitar a leitura.	
Exemplo Detalhado Nesse exemplo, existe apenas um caso de uso chamado <i>Registrar Proposta de Seguro</i> . Nesse caso de uso, o Corretor de Seguros irá registrar todas as informações sobre o segurado, os beneficiários e assim por diante. Quando todas as informações são informadas pelo Corretor de Seguros, o caso de uso checa se a proposta pode ser enviada imediatamente para aprovação ou se uma avaliação mais completa da proposta deve ser feita. O fluxo básico fornece uma visão geral de como normalmente o caso de uso é executado. A descrição inclui referências para outras subseções que contém descrições mais detalhadas de partes do fluxo.	
Exemplo: Caso de uso <i>Registrar Proposta de Seguro</i> Descrição Resumida: O Corretor de Seguros envia uma proposta de seguro de vida. O caso de uso registra a proposta no Diário e informa o Auditor de aprovação ou o Comitê de aprovação sobre a nova proposta. Ator: Corretor de Seguros Fluxo Básico: 1. O caso de uso inicia quando o Corretor de Seguros escolhe registrar uma proposta para um novo seguro de vida. 2. O caso de uso pergunta se é uma nova proposta ou um rascunho da proposta já existe. 3. O Corretor de Seguros informa a identificação de um rascunho de proposta que já existe ou seleciona criar uma nova. 4. Se uma identificação é informada, o caso de uso busca o rascunho da proposta, de outra forma, uma nova proposta é criada e sua identificação e apresentada para o Corretor de Seguros. 5. O caso de uso apresenta informações de nome, endereço, etc. que podem ser modificadas pelo Corretor de Seguros (Ver Seção Registrar Nome de Segurado) 6. O caso de uso apresenta informação de Status que pode ser modificado pelo Corretor de Seguros (Ver Seção Registrar Status do Segurado)	

7. O caso de uso apresenta informações sobre os beneficiários que podem ser modificadas pelo Corretor de Seguros (Ver Seção Registrar Beneficiários)
8. O caso de uso calcula o valor preliminar a ser pago baseado nas informações do Corretor de Seguros (Ver Seção Calcular Valor Preliminar) e apresenta esse valor.
9. O caso de uso apresenta informações sobre a forma de pagamento que pode ser modificado pelo Corretor de Seguros (Ver Seção Registrar Forma de Pagamento)
10. O caso de uso mostra as informações armazenadas sobre a proposta para o Corretor de Seguros e solicita a aceitação da proposta.
11. O Corretor de Seguros aceita, aguarda aceitação futura ou remove a proposta:
12. Se o Corretor de Seguros escolhe remover a proposta, todas as informações referentes a proposta são removidas e o caso de uso é encerrado.
13. Se o Corretor de Seguros escolhe aceitar futuramente a proposta, o caso de uso é encerrado.
14. Se o Corretor de Seguros aceita a proposta, ela é enviada para o Sistema de Diário.
15. O caso de uso verifica se os valores relacionados a proposta do segurado estão dentro dos limites de aprovação. Caso sim, o status da proposta é modificado para “Aprovado” e uma notificação é enviada para o Auditor de Aprovação. Se alguns dos valores da proposta do segurado estão fora dos limites, o status da proposta é modificado para “Em avaliação” e uma notificação é enviada para o Comitê de Aprovação.
16. O caso de uso termina.

Fluxos Alternativos:

No passo 3 – Não existe o número de identificação registrado no sistema. O caso de uso irá perguntar ao Corretor de Seguros para fornecer novamente um número de identificação ou escolher criar uma nova proposta. O caso de uso continua onde o Corretor de Seguros fornece a identificação da proposta.

SubFluxos (Subseções):

Registrar Nome de Segurado:

[Descrição do sub-fluxo]

Registrar Status do Segurado:

[Descrição do sub-fluxo]

Seção Registrar Beneficiários:

[Descrição do sub-fluxo]

Seção Registrar Forma de Pagamento:

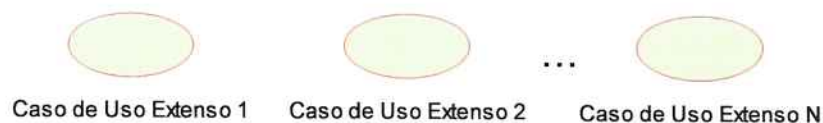
[Descrição do sub-fluxo]

Tabela 3.12 – Padrão *Múltiplos Fluxos*.

Padrão: <i>Múltiplos Fluxos</i>	Categoria: <i>Templates</i>
Parte do caso de uso afetada: Modelo	
<p>Discussão: Quando duas ou mais variações (variantes), igualmente importantes, podem ser observadas em um caso de uso, é possível dividir o caso de uso, promovendo as variações em novos casos de uso. É importante lembrar que não é uma boa idéia separar pequenos fluxos alternativos ou fluxos de exceção em novos casos de uso, pois a leitura de um modelo como um todo pode ficar mais complicada. Se um pequeno fluxo alternativo é usado em vários casos de uso, ele pode ser extraído para um caso de uso do tipo inclusão (<i>Include</i>). Similarmente, se existem sentenças, normas ou regras associadas ao negócio que são válidas para muitos casos</p>	

de uso elas podem ser referenciadas como regras de negócio nas descrições dos casos de uso. Um estudo do padrão *Regras de Negócio* é interessante.

Modelo:



Aplicabilidade: Esse padrão é aplicável para casos de uso compostos de vários fluxos alternativos. Nesse caso, cada fluxo alternativo pode ser modelado como um caso de uso separado. Essa separação irá manter uma descrição mais curta para cada caso de uso e, conseqüentemente, mais simples de ser gerenciado.

Exemplo: Não apresentado.

3.3.6 Grupo – Visões Ortogonais

Tabela 3.13 – Padrão *Especialização*

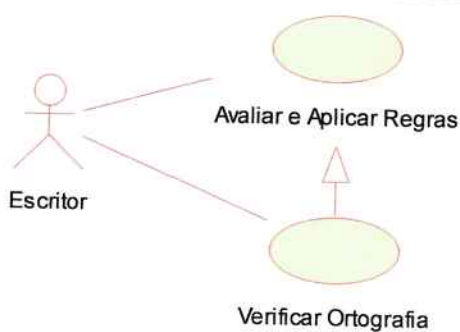
<i>Padrão: Especialização</i>	<i>Categoria: Templates</i>
Parte do caso de uso afetada: Modelo	
<p>Discussão: Em alguns sistemas, como por exemplo, em sistemas baseados em regras, a visão externa do sistema é completamente diferente da visão interna. Dessa forma, o que o usuário percebe pode ser completamente diferente do que realmente está desenvolvido. Isso causa um problema imediato para o desenvolvimento, pois diferentes <i>Stakeholders</i> terão uma percepção diferente de como o sistema funciona.</p> <p>Um editor de textos possui ferramentas para verificação de ortografia, gramática e inclusão de novas palavras. Um usuário do editor de textos provavelmente irá modelar essas ferramentas como três casos de uso diferentes.</p> <p>Entretanto, pode-se assumir que essas três ferramentas são uma parte do editor de texto que são implementadas utilizando uma técnica de aplicação de regras. Cada uma das regras possui um <i>Template</i> e um corpo. O <i>Template</i> é utilizado para comparação de diferentes partes do texto escrito. Se o <i>Template</i> coincide com a parte do texto analisada, o corpo da regra é executado. Essa abordagem pode ser percebida por um desenvolvedor, ou seja, os usos das ferramentas de verificação de ortografia, gramática e inclusão de novas palavras são usos específicos do mecanismo de checagem de regras.</p> <p>Considerando que existem dois pontos de vista diferentes sobre o uso do sistema, fica inviável a utilização de apenas um caso de uso para representar o fluxo de ações. Dessa forma, é altamente recomendável a aplicação de relacionamentos de generalização e especialização no modelo.</p>	

Modelo:

Aplicabilidade: Esse padrão é aplicável quando o modelo de casos de uso deve mostrar que as diferentes utilizações do sistema são casos especiais do uso geral do sistema, e que todas elas são realizadas do mesmo modo internamente, mas percebidas diferentemente por usuários diferentes.

Exemplo:

Esse exemplo mostra a descrição dos casos de uso *Avaliar e Aplicar Regras* e *Verificar Ortografia* conforme mencionados na descrição desse padrão.

**Caso de uso:** *Verificar Ortografia*

Descrição resumida: O caso de uso verifica a ortografia e apresenta qualquer erro para o escritor que pode escolher corrigi-los.

Ator: Escritor

Fluxo básico:

O caso de uso inicia quando o escritor escolhe verificar o ortografia do texto.

Avaliação e Aplicação de Regras

1. O caso de uso inicia a partir do início do texto e verifica se cada palavra encontra-se no dicionário.
2. O Escritor pode escolher a qualquer momento cancelar a verificação e nesse caso o caso de uso termina.
3. Se a palavra é encontrada no dicionário, o caso de uso continua com a próxima palavra.
4. Se a palavra não é encontrada no dicionário, o caso de uso apresenta a palavra errada para o Escritor e solicita que seja selecionada uma das seguintes opções:
 - a. Se o Escritor fornece uma nova palavra, a palavra que não foi interpretada é

<p>substituída e a verificação contínua.</p> <p>b. O Escritor pode escolher continuar a verificação sem fornecer uma nova palavra, e nesse caso a palavra não interpretada permanece sem alterações.</p> <p>c. O Escritor pode escolher adicionar a palavra para o dicionário. O caso de uso adiciona a palavra selecionada para o dicionário e continua a verificação com a próxima palavra.</p> <p><u>Final de Fluxo</u></p> <p>O caso de uso termina como descrito em Avaliar e Aplicar Regras</p>

Tabela 3.14 – Padrão *Descrição*

<i>Padrão: Descrição</i>	<i>Categoria: Templates</i>
Parte do caso de uso afetada: Descrição	
<p>Discussão: Em alguns sistemas, como por exemplo, em sistemas baseados em regras, a visão externa do sistema é completamente diferente da visão interna. Dessa forma, o que o usuário percebe pode ser completamente diferente do que realmente está desenvolvido. Isso causa um problema imediato para o desenvolvimento, pois diferentes <i>Stakeholders</i> terão uma percepção diferente de como o sistema funciona.</p> <p>O fluxo de ações que acontece no sistema, do ponto de vista de um <i>Stakeholder</i> interno, pode ser muito maior que o fluxo de ações que um <i>Stakeholder</i> externo percebe. Dessa forma, um grupo de leitores do caso de uso poderá reclamar sobre a descrição de ações que não são relevantes do ponto de vista deles, enquanto outros grupos de leitores sentirão necessidade de um maior detalhamento. Para minimizar o risco de ter duas descrições do mesmo caso de uso, que não são compatíveis, e ao mesmo tempo evitar que o número de documentos relacionados aos casos de uso aumente, descrições adicionais podem ser colocadas em subseções, separadas da descrição dos casos de uso.</p>	
<p>Aplicabilidade: Esse padrão pode ser usado quando <i>Stakeholders</i> diferentes têm diferentes percepções do caso de uso, necessitando, assim, de níveis de detalhamento diferentes do mesmo caso de uso.</p>	
<p>Exemplo:</p> <p>Suponha-se o caso de uso para uma chamada telefônica. A maioria dos <i>Stakeholders</i> não sabe o que acontece dentro da rede de telefonia enquanto uma ligação é realizada. Certamente, a maior parte dos interessados nesse caso de uso realmente não precisa entender detalhes do funcionamento da rede. Dessa forma, o trabalho de leitura desse caso de uso para esses <i>Stakeholders</i> é algo complicado e desnecessário. Entretanto, para um conjunto de desenvolvedores é absolutamente importante entender o que acontece na rede durante uma ligação.</p> <p>Suponha que o fluxo de ações que também são importantes para os desenvolvedores seja relacionado a <i>Encontrar Rota e Transmitir o ID do Assinante Chamador</i>.</p> <p>Para uma visão externa do caso de uso <i>Realizar Chamada</i>, podem-se definir as seguintes seções:</p> <ol style="list-style-type: none"> 1 – Introdução 2 – Pontos de Extensão 3 – Fluxo Básico 	

4 – Fluxos Alternativos

Para uma visão interna do caso de uso *Realizar Chamada*, pode-se definir as seguintes seções:

- 1 – Introdução
- 2 – Pontos de Extensão
- 3 – Fluxo Básico
 - 3.1 Encontrar Rota
 - 3.2 Transmitir o ID do Assinante Chamador
- 4 – Fluxos Alternativos

Como resultado final, o caso de uso *Realizar Chamada*, pode ser definido, com os dois níveis de abstração criando-se a seção “Descrições Adicionais” da seguinte maneira:

- 1 – Introdução
- 2 – Pontos de Extensão
- 3 – Fluxo Básico
- 4 – Fluxos Alternativos
- 5 – Descrições Adicionais
 - 5.1 Encontrar Rota
 - 5.2 Transmitir o ID do Assinante Chamador

3.3.7 Grupo – Seqüência de Casos de Uso

Tabela 3.15 – Padrão *Seqüência de casos de uso*

Padrão: <i>Seqüência de casos de uso</i>	Categoria: <i>Templates</i>
Parte do caso de uso afetada: Descrição	
<p>Discussão: Em muitos sistemas, o usuário deve realizar algum tipo de procedimento inicial antes que alguns serviços do sistema estejam disponíveis (por exemplo, <i>Log In</i>). Depois de usar alguns desses serviços, o usuário deve executar algum procedimento de encerramento (por exemplo, <i>Log Out</i>). Normalmente, a maioria desses serviços não relacionam-se uns aos outros e não estão disponíveis ao mesmo tempo. O Usuário terá que navegar através de um menu hierárquico ou por um conjunto de páginas <i>Web</i> para encontrar os serviços desejados e estar apto a iniciar o caso de uso correspondente.</p> <p>Em um <i>Internet Bank</i>, o cliente inicia uma página <i>Web</i> que requisita basicamente o ID do cliente e senha de acesso correspondente. Uma vez que essas informações são validadas, o sistema apresenta uma página de entrada com algumas notícias do banco e algum tipo de informação personalizada. Nessa página o cliente pode escolher entre grupos de serviços, como por exemplo, verificar saldos e extratos e fazer algum tipo de aplicação. Quando o cliente solicita algum desses grupos, uma nova página <i>Web</i> é apresentada que lista informações mais detalhadas. O cliente pode então escolher um serviço chamado <i>Listar Transações do Último Mês</i> para uma determinada conta.</p>	

Nesse exemplo, o caso de uso *Mostrar Últimas Transações do Mês*, recebe um número de conta como entrada e busca todas as transações realizadas nos últimos cinco dias do mês para a conta e apresenta para o usuário. Do ponto de vista funcional, esse caso de uso pode ser realizado a qualquer momento, tão logo que um número de conta seja fornecido.

No sistema, a interface restringe que serviços podem ser selecionados em determinados níveis. Existem duas maneiras de expressar que determinado caso de uso seja chamado em uma ordem específica: através de pré-condições e/ou máquinas de estado.

A pré-condição de um caso de uso é usada para indicar qual estado o sistema deve estar para que o caso de uso seja iniciado. Especificadores de interface de usuário e programadores, dentre outros, podem usar a pré-condição como entrada para seus trabalhos. Uma máquina de estados pode ser usada para especificar quais são os diferentes estados do sistema e quais transações podem ser realizadas entre os estados. Como complemento, pode-se usar também um diagrama de atividades para mostrar a ordem em que os casos de uso devem ser chamados.

Relacionamentos de Inclusão e Extensão não devem ser usados para expressar ordem de execução dos casos de uso. De maneira geral, relacionamentos são usados para adicionar um conjunto de ações dentro de outros casos de uso. A única situação onde relacionamentos são utilizados é quando um caso de uso faz uso de um valor calculado por outro caso de uso. Dessa forma, essa situação define uma ordem temporal entre as ações descritas nos diferentes casos de uso.

Aplicabilidade: É aplicável quando existe uma ordenação temporal entre casos de uso mas nenhuma informação retornada por uma instância do caso de uso será incluída na instância do outro casos de uso.

Exemplo detalhado:

Esse exemplo utiliza pré-condições para especificar a ordem dos casos de uso. São descritos quatro casos de uso de um sistema de *Internet Bank*: *Mostrar Página de Navegação*, *Mostrar Contas*, *Pagar Faturas* e *Mostrar Últimas Transações do Mês*. O primeiro caso de uso é usado para navegação entre os diferentes serviços do banco que são apresentados em diferentes páginas. O caso de uso *Mostrar Contas* pode somente ser executado quando a página de serviços é mostrada. O mesmo é válido para o caso de uso *Pagar Faturas*. O último caso de uso apresenta as transações realizadas por uma conta nos últimos cinco dias do mês. Esse caso de uso pode ser executado somente se uma conta for apresentada na página Web corrente.

Exemplo:

Caso de uso: *Mostrar Página de Navegação*

Descrição Resumida: O Cliente Internet seleciona um dos serviços oferecidos e o caso de uso mostra a página *web* correspondente ao caso de uso selecionado.

Ator: Cliente Internet

Pré-Condição: Uma página *Web* deve ser apresentada para o Cliente Internet de onde a seleção do serviço pode ser feito.

Fluxo básico

1. O caso de uso inicia quando o Cliente Internet seleciona um dos serviços que são apresentados.
2. O caso de uso busca o conteúdo do serviço selecionado, isto é, a descrição do serviço e que casos de uso devem ser iniciados a partir desse serviço.
3. O caso de uso mostra o conteúdo para o Cliente Internet usando a página web conectada para o serviço selecionado.
4. O caso de uso termina.

Caso de uso: *Mostrar Contas*

Descrição Resumida: O caso de uso apresenta as contas do Cliente Internet

Pré-Condição: Uma página *Web* que inclui a opção de ver as contas de cliente deve ser apresentada para o Cliente Internet

Fluxo Básico

1. O caso de uso inicia quando o Cliente Internet seleciona ver as contas registradas para o ator.
2. O caso de uso armazena a página web de onde a seleção foi feita.
3. O caso de uso identifica o Cliente Internet para buscar todas as contas disponíveis para o cliente. Para cada uma dessas contas, o caso de uso busca o tipo da conta, o número e o balanço da conta. Essas informações são apresentadas para o Cliente Internet em uma nova página web.
4. Quando o Cliente Internet escolhe encerrar a visualização das contas, o caso de uso restaura a página anteriormente armazenada e apresenta para o Cliente Internet.
5. O caso de uso termina.

Caso de uso: *Pagar Faturas*

Descrição Resumida: O caso de uso registra faturas para pagamento fornecidas pelo Cliente Internet

Pré-Condição: Uma página web que inclui a opção de registrar contas deve ser apresentada para o Cliente Internet

Fluxo Básico:

1. O caso de uso inicia quando o Cliente Internet seleciona pagar uma coleção de faturas.
2. O caso de uso armazena a página que a seleção foi feita.
3. O caso de uso identifica o Cliente Internet e busca a conta corrente do Cliente e continua realizando o seguinte laço até que o Cliente Internet encerre o registro das faturas:
 - a. O caso de uso apresenta uma página web para o Cliente Internet com o número da conta corrente e o balanço.
 - b. O Cliente Internet é solicitado para entrar com o valor a ser pago, a conta de destino, a data que a fatura deverá ser paga e uma mensagem de transação.
 - c. O Cliente Internet entra com as informações requisitadas que são armazenadas pelo caso de uso junto ao número da conta corrente do Cliente Internet.
4. Quando o Cliente Internet escolhe interromper o registro de faturas para serem pagas, o caso de uso restaura a página anteriormente armazenada e apresenta para o Cliente Internet.
5. O caso de uso termina.

Fluxos Alternativos:

- No Passo 1 - O Cliente Internet possui apenas conta poupança: o caso de uso apresenta uma mensagem de erro informando que esse serviço não está disponível para o Cliente Internet. O caso de uso é encerrado.

Caso de Uso: *Mostrar Últimas Transações do Mês*

Descrição Resumida: O caso de uso apresenta todas as transações dos últimos cinco dias do mês corrente para uma conta de um Cliente Internet

Pré-Condição: Uma página web que inclua o número da conta deve ser apresentada para o Cliente Internet.

Fluxo Básico:

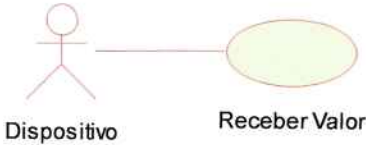
1. O caso de uso inicia quando o Cliente Internet seleciona ver as transações de uma conta ao mesmo tempo em que fornece o número de uma conta.
2. O caso de uso armazena a página web de onde a seleção foi feita.
3. O caso de uso utiliza o número da conta para buscar todas as transações realizadas nos últimos cinco dias do mês.
4. Para cada uma das transações, são mostradas para o Cliente Internet em uma nova página web : A data, o montante, se foi débito ou crédito e o texto de transação.
5. Quando o Cliente Internet escolhe ver o resumo de transações, o caso de uso busca a página anterior e apresenta para o Cliente Internet.
6. O caso de uso termina.

Fluxos Alternativos: Não existem.

3.4 Catálogo de padrões – Domínio Específico

3.4.1 Grupo – Meio Externo Passivo

Tabela 3.16 – Padrão *Meio Externo Passivo*

Padrão: <i>Meio Externo Passivo</i>	Categoria: <i>Domínio Específico</i>
Parte do caso de uso afetada: Modelo	
<p>Discussão: Em alguns domínios é bastante comum que dispositivos forneçam algum tipo de informação para o sistema. Um dispositivo pode ser, por exemplo, um termômetro que transmite informações sobre a temperatura corrente de um líquido ou um altímetro que continuamente transfere informações sobre altitudes relativas ao nível do mar. Em situações como essas, são coletadas propriedades de meios externos ao sistema, nesse exemplo, o líquido e o ar respectivamente. Nenhum desses meios, por si só, envia algum tipo de estímulo para o sistema. Por essa razão, não seria natural dizer que o meio é um ator no sistema e que o meio envia informações para o sistema. Também seria errado dizer que o sistema envia informações para o meio.</p> <p>Dessa forma, quando uma entidade externa ao sistema é passiva, possivelmente meios abertos como líquido e gás, a modelagem é feita utilizando-se um dispositivo de medição como sendo um ator para o sistema. O dispositivo pode interagir ativamente com o meio e coletar informações sobre ele.</p> <p>É importante enfatizar que, geralmente, dispositivos não são modelados como atores, mas sim os papéis relacionados aos indivíduos, que estão por de trás dos dispositivos. Essa situação de meios externos passivos é exceção.</p>	
Aplicabilidade:	
<p>Esse padrão é aplicado quando não existe um mapeamento direto de um usuário que envia informações de um meio externo para o sistema. Nesse padrão, um dispositivo deve examinar o ambiente passivo e enviar informações para o sistema.</p>	
Modelo:	
	
Exemplo Detalhado:	
<p>Suponha um sistema para controle de aquecimento central. O modelo consiste de dois casos de uso para controle do aquecedor. O primeiro caso de uso lê valores de entrada e avalia a variação de temperatura. Se a temperatura difere mais do que dois</p>	

graus da temperatura normal, o caso de uso irá ligar ou desligar o aquecedor e a potência utilizada nesse caso será a metade da capacidade do aquecedor (considerando o aquecedor ligado). Se o valor de entrada estiver cinco graus abaixo da temperatura normal, o segundo caso de uso irá iniciar e será utilizado a potência máxima do aquecedor por um minuto. Depois disso, o aquecedor é desligado e o caso de uso é encerrado.

Nesse exemplo, o aquecedor é considerado parte do sistema e dessa forma não é modelado como um ator. Se ele não fosse parte do sistema poderia ser modelado como um ator e relacionar-se com os dois casos de uso.

Exemplo:

Caso de uso: *Ligar e Desligar Aquecedor*

Descrição resumida: O caso de uso recebe um valor de entrada do aquecedor e dependendo do valor ele pode ligar ou desligar o aquecedor.

Fluxo Básico:

O caso de uso tem dois fluxos básicos diferentes:

- Ligar o aquecedor
- Desligar o aquecedor

Ligar o aquecedor

1. O caso de uso inicia quando o Termômetro fornece um valor de entrada.
2. Se o valor está até dois graus abaixo do valor desejado e o aquecedor não está ligado, o caso de uso liga o aquecedor com funcionamento a meia potência.
3. O caso de uso termina.

Desligar o aquecedor

1. O caso de uso inicia quando o termômetro fornece um valor de entrada.
2. Se o valor está mais do que dois graus acima do valor desejado e o aquecedor não está ligado, o caso de uso desliga o aquecedor.
3. O caso de uso termina.

Caso de uso: *Ligar aquecedor com potência total*

Descrição resumida: Quando o caso de uso recebe um valor de entrada bastante baixo do termômetro, o caso de uso liga o aquecedor com potência máxima por um minuto.

Fluxo Básico:

1. O caso de uso inicia quando o termômetro fornece um valor de entrada.
2. Se o valor está pelo menos cinco graus a baixo do valor desejado, o caso de uso liga o aquecedor com potência máxima.
3. Depois de um minuto, o caso de uso desliga o aquecedor.
4. O caso de uso termina.

3.4.2 Grupo – Controle de Acesso

A discussão apresentada a seguir é aplicável para todos os padrões apresentados nesse grupo. Dessa forma, ela não será repetida nas tabelas dos padrões específicos.

Na maioria dos grandes sistemas, os usuários não têm acesso para realizar qualquer operação (podem-se considerar exceção, os usuários Administradores). Para proteger dados, recursos e funções de operações indevidas, um sistema deve possuir alguma facilidade para controle de acesso. O nível de sofisticação do mecanismo de proteção pode variar entre os sistemas. Os padrões apresentados nesse trabalho para controle de acesso focam-se na leitura e atualização de informações e uso de funções, em uma abordagem simples. Entretanto, da perspectiva dos casos de uso, essa abordagem pode ser estendida com conceitos adicionais.

Suponha-se o caso de uma biblioteca pública, onde todos os visitantes podem acessar terminais para procurar informações sobre livros ou revistas, a partir dos títulos e autores e, além disso, verificar a sua disponibilidade para empréstimos. Os usuários visitantes não podem fazer alterações nos dados cadastrais de um livro ou cadastrar novas informações; somente os bibliotecários têm acesso para realizar esses tipos de operações.

Em cada caso de uso onde o controle de segurança é necessário, deve-se incluir uma checagem de direitos de acesso, logo que o sistema acesse alguma informação ou função restrita. Essa checagem pode ser diretamente expressa em cada caso de uso relevante. Nesse caso, pode-se aplicar o padrão *Controle de Acesso: Checagem Embutida*.

Caso a checagem afete vários casos de usos, pode-se criar um caso de uso abstrato para essa checagem que deve ser incluído em todos os casos de uso que necessitem realizar a checagem dos acessos. Nessa abordagem, aplica-se o padrão *Controle de Acesso: Checagem Explícita*. A vantagem desse modelo é a separação do controle de acesso em um caso de uso da parte que fica explícita no modelo, podendo então ser revisado separadamente. A desvantagem é que o caso de uso de inclusão, algumas vezes, precisa ser descrito em um nível de abstração muito baixo, se comparado aos outros casos de uso o modelo. Nesse caso, pode-se aplicar o padrão *Checagem Embutida* descrito anteriormente e utilizar também o padrão *Regras de Negócio* considerando o controle de acesso descrito no documento de regras de negócio.

O gerenciamento dos direitos de acesso dos diferentes usuários também pode ser expresso no modelo de casos de usos. O caso de uso *Gerenciar Direitos de Acesso*, iniciado pelo ator Administrador de Sistemas é um exemplo típico de um caso de uso do tipo CRUD. Ele modela como os direitos de acesso são criados, modificados ou removidos.

Uma solução mais avançada para segurança é expressa pelo padrão *Controle de acesso: Unidades de Segurança Dinâmica*. Ele inclui um caso de uso adicional chamado *Definir Unidade de Segurança*, que compreende conceitos de unidades para que direitos de acessos possam ser definidos para usuários individuais. Nesse padrão,

é possível dinamicamente definir quais funções, recursos ou partes de informações deverão ser incluídos em um direito de acesso. Por exemplo, a direção da biblioteca pública relatou que muitos visitantes, depois de realizarem uma busca no sistema, enviam as informações de um livro para a impressora simplesmente para saber onde localizá-lo na biblioteca. O uso dessa impressora representa um montante significativo de dinheiro, que poderia ser utilizado para outros propósitos. Nesse exemplo, o Administrador de Sistemas poderia definir a impressora como um novo tipo de unidade de segurança e liberar para cada bibliotecário o direito de acessá-la.

Em muitos sistemas, o Administrador de Sistemas não está envolvido, quando muitos direitos de acesso são definidos. Os direitos de acesso são definidos a partir de valores padrões e fornecidos assim que uma nova parte de informação é criada ou uma nova função é adicionada. Por exemplo, quando se cria um novo arquivo no sistema, ele recebe direitos de acesso padrões que normalmente são herdados do diretório onde foi criado. Nesse cenário, pode-se utilizar o padrão *Controle de Acesso: Atribuição Interna*.

Tabela 3.17 – Padrão *Checagem Embutida*

Padrão: <i>Checagem Embutida</i>	Categoria: <i>Domínio Específico</i>
Parte do caso de uso afetada: Modelo	
Discussão: conforme a discussão geral do grupo	
Aplicabilidade: Esse padrão pode ser utilizado quando não existe a necessidade de expressar a checagem dos direitos de acesso explicitamente no modelo.	
Modelo:	
 <pre> graph LR A[Administrador de Sistemas] --- UC((Gerenciar Direitos de Acesso)) </pre>	
Exemplo: Exemplo para o grupo (Tabela 3.21)	

Tabela 3.18 – Padrão *Checação Explícita*

Padrão: <i>Checação Explícita</i>	Categoria: <i>Domínio Específico</i>
Parte do caso de uso afetada: Modelo	
Discussão: conforme discussão geral do grupo	
Aplicabilidade: Esse padrão deve ser aplicado quando existe a necessidade de expressar a checagem de direitos de acesso, explicitamente, no modelo.	
Modelo:	
<pre> graph LR Admin[Administrador de Sistemas] --- Gerenciar[Gerenciar Direitos de Acesso] Gerenciar --- Checar[Checar Direitos de Acesso] Requisicao[Caso de Uso para Requisição de Recurso] -.-> <<include>> Checar </pre>	
Exemplo: Exemplo para o grupo (Tabela 3.21)	

Tabela 3.19 – Padrão *Unidades de Segurança Dinâmica*

Padrão: <i>Unidades de Segurança Dinâmica</i>	Categoria: <i>Domínio Específico</i>
Parte do caso de uso afetada: Modelo	
Discussão: conforme discussão geral do grupo.	
Aplicabilidade: Esse padrão pode ser utilizado quando existe a necessidade de checagem de direitos de acesso por recursos do sistema. Além disso, o gerenciamento dos acessos deve ser feito dinamicamente com o sistema em execução.	
Modelo:	
<pre> graph LR Admin[Administrador de Sistemas] --- Gerenciar[Gerenciar Direitos de Acesso] Admin --- Seguranca[Gerenciar Segurança da Unidade] </pre>	
Exemplo: Exemplo para o grupo (Tabela 3.21)	

Tabela 3.20 – Padrão *Atribuição Interna*.

Padrão: <i>Atribuição Interna</i>	Categoria: <i>Domínio Específico</i>
Parte do caso de uso afetada: Modelo	
Discussão: conforme discussão geral do grupo.	
Aplicabilidade: Pode ser aplicado quando a criação de recursos do sistema é dinâmica.	
Modelo:	
<pre> graph TD Actor[Administrador de Sistemas] --- UC1[Gerenciar Direitos de Acesso] UC2[Caso de Uso para Criação de Recurso] -.-> <<include>> UC3[Configurar Direitos de Acesso] UC4[Caso de Uso para Requisição de Recurso] -.-> <<include>> UC5[Checar Direitos de Acesso] </pre>	
Exemplo: Exemplo para o grupo (Tabela 3.21)	

A tabela 3.21 mostra o exemplo aplicável para todo o grupo de padrões – *Controle de Acesso*

Tabela 3.21 – Exemplo para o grupo de padrões – *Controle de Acesso***Exemplo detalhado:**

Nesse exemplo, é apresentada uma parte de um sistema para controle de conferências. Participantes podem usar o sistema para verificar e registrar em diferentes eventos. O usuário deverá estar autenticado no sistema para poder utilizá-lo. Cada participante pode ler informações em formulários públicos ou páginas *Web*. Se o acesso a um formulário é restrito, somente alguns participantes poderão ler o conteúdo e somente um subconjunto predefinido de participantes poderão registrar no evento correspondente. Um Administrador é responsável por criar e atualizar os formulários assim como definir quem pode ler e se inscrever.

Exemplo:**Caso de uso:** *Gerenciar Formulário*

Descrição Resumida: O Administrador de Formulários utiliza o caso de uso para criar e atualizar formulários.

Ator: Administrador de Formulários

Fluxo Básico:

1. O caso de uso inicia quando o Administrador de Formulários escolhe atualizar um formulário existente ou criar um novo formulário.

2. O caso de uso requisita ao Administrador de Formulários que seja informado o usuário e a senha.
3. O Administrador de Formulários fornece as informações requeridas.
4. O caso de uso verifica se o usuário e senha fornecidos estão corretos e se o usuário tem permissão para modificar formulários.
5. O caso de uso busca os nomes dos formulários existentes, apresenta para o Administrador de Formulários e requisita para que um deles seja modificado ou se um novo deverá ser criado.
6. O Administrador de Formulários seleciona um formulário.
7. O caso de uso busca e apresenta o formulário juntamente ao seu nome. Se ao invés disso, o Administrador de Formulários seleciona criar um novo formulário, o caso de uso apresenta um formulário vazio.
8. O Administrador de Formulário entra ou modifica as informações dentro do formulário ou modifica o nome do formulário.
9. O Administrador de Formulário salva o formulário novo ou modificado. O caso de uso verifica se o nome não está vazio e armazena o formulário no sistema.
10. O caso de uso requisita para o Administrador de Sistema se o formulário deverá ter controle de acesso. Se afirmativo, o formulário é marcado como uma unidade de segurança. Caso contrário, o formulário não é marcado como uma unidade de segurança (permanece como está no momento). O caso de uso armazena as informações.
11. O caso de uso termina.

Fluxos Alternativos:

- No passo 1 - O Administrador de Formulários não possui autoridade suficiente para modificar ou criar novos formulários: o Administrador de Formulários é notificado e o caso de uso é encerrado.

Caso de uso: *Gerenciar Direitos de Acesso de um Formulário*

Descrição Resumida: O Administrador de Formulário utiliza o caso de uso para gerenciar os direitos de acesso dos participantes para os formulários.

Ator: Administrador de Formulários

Fluxo Básico:

1. O caso de uso inicia quando o Administrador de formulários escolhe atribuir direitos de acesso para um formulário.
2. O caso de uso requisita ao Administrador de Formulários que seja informado o usuário e a senha.
3. O Administrador de Formulários fornece as informações requeridas.
4. O caso de uso verifica se o usuário e senha fornecidos estão corretos e se o usuário tem permissão para modificar direitos de acesso nos formulários.
5. O caso de uso busca o nome dos formulários existentes que estão marcados como unidade de segurança, apresenta para o Administrador de Formulários e solicita que um deles seja selecionado para ser atualizado.
6. O Administrador de Formulários seleciona um formulário.
7. O caso de uso busca e apresenta o formulário juntamente ao seu nome.
8. O caso de uso busca a lista de participantes que tem qualquer acesso ao form e apresenta-os para o Administrador de Formulários.
9. O caso de uso requisita ao Administrador de Formulários que seja fornecido alterações na lista apresentada, ou seja, quais participantes deverão ser adicionados ou removidos e para cada um deles, que tipo de direito será atribuído (Ler um Form e/ou Registrar).
10. O Administrador de Formulários fornece as alterações requisitadas e o caso de uso atualiza a lista.
11. O caso de uso termina.

Fluxos Alternativos:

Nível de acesso insuficiente

No passo 1 - O Administrador de Formulários não possui autoridade suficiente para modificar os direitos de acesso de um formulário: o Administrador de Formulários é notificado e o caso de uso é encerrado.

Caso de uso: *Verificar Direitos de Acesso*

Descrição Resumida: O caso de uso descreve como um direito de acesso de um participante para um formulário é realizado. O caso de uso é abstrato.

Fluxo Básico:

Quando esse caso de uso é incluído por outro caso de uso, a instância do caso de uso continua realizando as seguintes ações:

1. O caso de uso verifica se o formulário é marcado como uma unidade de segurança:
 - a. Se o formulário é uma unidade de segurança, o caso de uso busca a identidade do usuário corrente e a usa junta com o identificador do formulário corrente para checar se o usuário tem o acesso para ler informações do formulário e se registrar através do formulário. O resultado das duas checagens constitui o resultado final da verificação de acesso.
 - b. Se o formulário não é uma unidade de segurança, um resultado positivo para leitura e registro constitui o resultado final da verificação de acesso.
2. O subfluxo termina e a instância continua como descrita no Caso de Uso Base depois de onde esse caso de uso foi incluído. O resultado da verificação fica disponível para o Caso de Uso Base.

Caso de uso: *Apresentar Informações do Evento*

Descrição Resumida: Um formulário contendo informações sobre um evento é apresentado para o Participante.

Pré-Condição: O Participante deverá estar autenticado

Fluxo Básico:

1. O caso de uso inicia quando o Participante escolhe ver as informações de um evento apresentadas em um formulário.
2. O caso de uso requisita qual formulário deverá ser apresentado.
3. O Participante entra com o nome do formulário. O Caso de Uso Incluído *Checar Direitos de Acesso* é executado.
4. Se o Participante tem acesso para ler o formulário, o caso de uso busca o formulário e o apresenta para o Participante.
5. Depois de ler o formulário selecionado, o Participante notifica o caso de uso.
6. O caso de uso fecha o formulário e é encerrado

Fluxos Alternativos:

No passo 1 - O Participante não possui direito de acesso para ver o conteúdo de um formulário: O Participante é notificado e o caso de uso é encerrado

Caso de uso: *Registrar para um Evento*

Descrição Resumida: Um formulário para registro a um evento é apresentado para um Participante.

Pré-Condição: O Participante deverá estar autenticado.

Fluxo Básico:

1. O caso de uso inicia quando o Participante escolhe se registrar para um evento.
2. O caso de uso requisita qual formulário deverá ser apresentado.
3. O Participante entra com o nome do formulário.
4. O Caso de Uso Incluído *Checar Direitos de Acesso* é executado.
5. Se o Participante tem acesso para ler o formulário, o caso de uso busca o formulário e o apresenta para o Participante. Se o Participante tem acesso para se registrar ao evento relacionado ao formulário, o caso de uso habilita a parte do formulário para registro.
6. Se o Participante seleciona ser registrado, o caso de uso armazena o identificador do usuário junto ao identificador do evento. De outra forma, nenhum registro é feito.
7. O caso de uso fecha o formulário e é finalizado.

Fluxos Alternativos:

Sem permissão para ver o conteúdo

No passo 1 - O Participante não possui direito de acesso para ver o conteúdo de um formulário: O Participante é notificado e o caso de uso encerra.

3.5 Conclusões

Os padrões de casos de uso apresentados no catálogo desse capítulo baseiam-se na idéia de *Design Patterns*. Entretanto, é importante destacar que existem diferenças fundamentais nos dois tipos de padrões. *Design Patterns* direcionam o desenho (projeto) do software, que produz um plano de implementação para um projeto de software. Padrões para casos de uso são – completamente - orientados para o ponto de vista dos atores e são considerados artefatos de reuso para engenheiros de requisitos.

4 ESTRATÉGIA PARA REVISÃO DE CASOS DE USO

Esse capítulo apresenta o detalhamento da Estratégia para Revisão de casos de uso, proposta principal desse trabalho, descrevendo como ela surgiu, sua estrutura e cenários aplicados ao SGP.

4.1 Introdução

O SGP, Sistema de Gerenciamento de Processos, foi um produto projetado para automatizar processos de negócios baseados em *Workflow*. Durante a entrega dos primeiros executáveis do software, constatou-se um grande número de defeitos provenientes de requisitos incorretos, incompletos ou desalinhados com as necessidades dos *Stakeholders*. Devido à gravidade dos problemas detectados, foi necessário revisar a configuração do processo de desenvolvimento do projeto do SGP.

Dentro da revisão do processo, a empresa direcionou um grande esforço na melhoria da qualidade dos modelos de casos de uso que eram gerados. O autor dessa monografia trabalhou fundamentalmente na criação de novos modelos de casos de uso e na revisão dos modelos que já tinham sido gerados em outras iterações de desenvolvimento.

Fazendo uma análise da visão do projeto do SGP e de outros artefatos que definiam as necessidades dos *Stakeholders*, ficou claro que os problemas começaram a partir da criação dos casos de uso, ou seja, as necessidades de negócio e de uso geral do sistema foram mapeadas corretamente na primeira fase do projeto, mas posteriormente, muitos modelos de caso de uso foram construídos erroneamente.

Antes de começar a revisão dos modelos propriamente dita, o primeiro passo do autor do trabalho foi um estudo profundo sobre os principais padrões para casos de uso. A decisão em usar padrões alinou-se às necessidades da empresa que buscava alternativas que auxiliassem na criação de modelos com alta produtividade e qualidade.

Nas pesquisas sobre padrões, foram encontradas dezenas de artigos sobre o assunto. Muitos desses padrões pesquisados, eram exatamente os mesmos, mas possuíam variações nos nomes, formatos e domínios de aplicabilidade. Buscando a convergência entre os padrões pesquisados, o autor da monografia baseou-se no trabalho de Overgaard e Palmkvist (2004). O trabalho dos autores em questão apresenta um catálogo de padrões que sintetizam boa parte dos padrões pesquisados.

A estratégia surgiu como consequência do trabalho de revisão de casos de uso que já estavam prontos. Dessa forma, considerando um processo de desenvolvimento, a estratégia pode ser aplicada logo que os primeiros modelos de casos de uso fiquem prontos e durante toda a etapa de levantamento de requisitos.

Durante a fase inicial de revisão dos modelos de casos de uso do SGP, foram identificadas soluções ruins que se repetiam nos diversos modelos. Essas soluções são consideradas antipadrões (*Anti-Patterns*). Como resultado final da Estratégia para Revisão, foi relacionada a aplicação de padrões de casos de uso para correção de antipadrões. Todos os exemplos que são apresentados na estratégia – que será descrita nas próximas seções desse texto – são situações reais e foram retirados do projeto do SGP.

Vale ressaltar que a estratégia foi montada a partir de um projeto para um sistema de BPM e aborda as falhas de modelagem mais comuns, que certamente ocorrem em qualquer projeto. Entretanto, seria impossível montar uma Estratégia para Revisão que cobrisse todos os aspectos possíveis de todos os projetos e domínios de negócio. O autor da monografia entende que esse é um trabalho inicial que pode ser complementado com trabalhos futuros, conforme sugerido no capítulo final.

4.2 Estrutura

A Estratégia para Revisão consiste em avaliar Situações que são identificadas em modelos de casos de uso. Para cada Situação deve ser analisado um conjunto de Tópicos, que podem ser aplicados no contexto da situação.

Um Tópico, de maneira geral, direciona a discussão e análise para um cenário real, baseado no SGP, que pode abordar o uso de antipadrões, e possíveis soluções, na maioria das vezes, baseadas no uso de padrões de casos de uso. Os Tópicos foram agrupados, conceitualmente, em Módulos.

A figura 4.1 mostra a estrutura que relaciona as Situações, Módulos/Tópicos e padrões que direcionam o uso da estratégia.

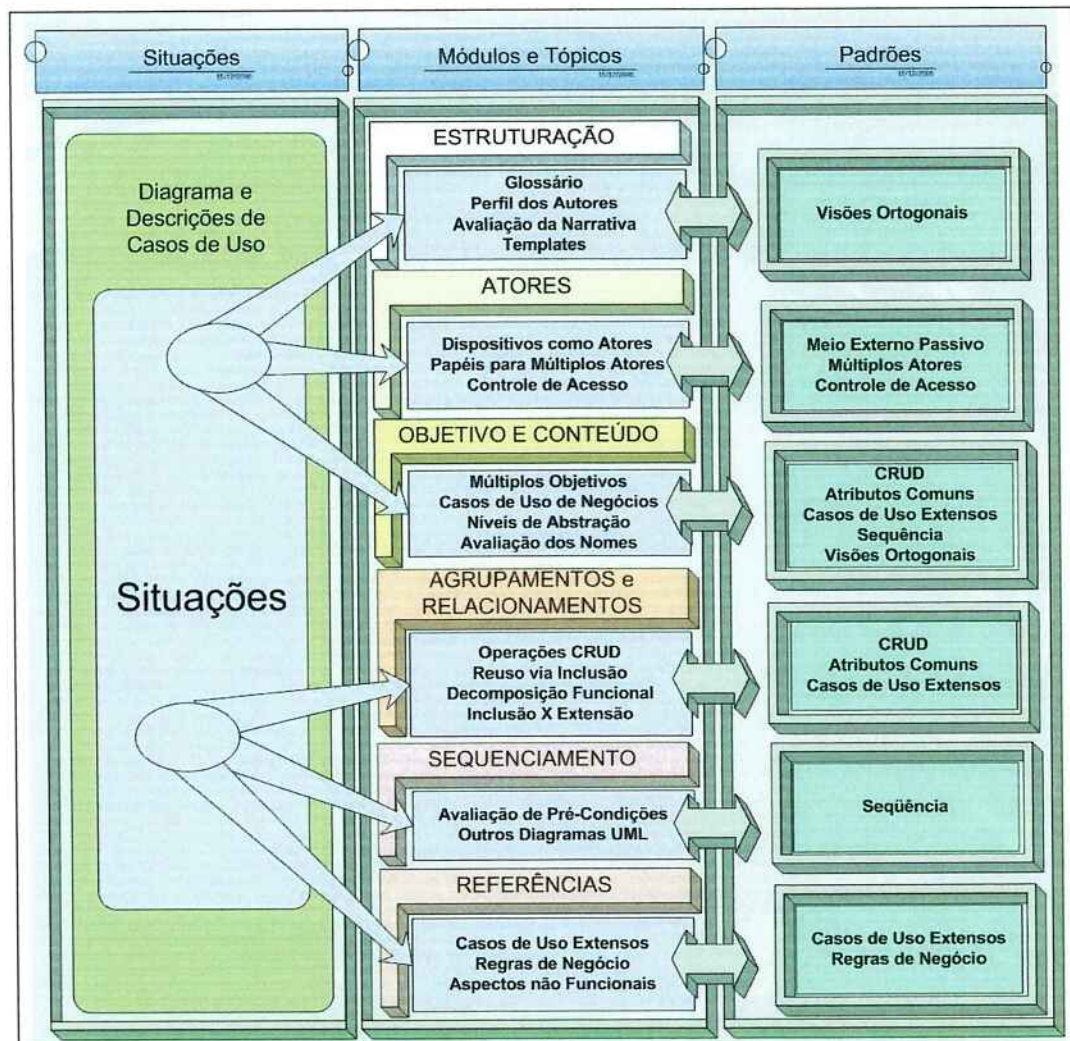


Fig. 4.1 – Estrutura de Módulos, Tópicos e Padrões da estratégia.

A Estratégia para Revisão é composta de 22 Situações que podem ser identificadas durante o processo de revisão. Cada Situação tem pelo menos um Tópico de revisão a ser considerado. A tabela 4.1 mostra a descrição de cada Situação e os Tópicos de revisão relacionados.

Tabela 4.1 – Descrição das Situações relacionadas aos Tópicos de revisão.

Id	Situações	Tópicos a serem analisados
1	Um caso de uso troca informações com um meio externo passivo, como, por exemplo, líquido e gás.	Dispositivos como Atores
2	Um caso de uso relaciona-se a múltiplos atores.	Papéis para Múltiplos Atores
3	Um ator inicia o fluxo de ações e recebe informações de um mesmo caso de uso.	Papéis para Múltiplos Atores
4	Um caso de uso relaciona-se a algum tipo de controle de acesso no sistema.	Controle de Acesso
5	O modelo de casos de uso ou a descrição do fluxo de ações é suficiente na visão de alguns <i>Stakeholders</i> , mas incompleto na visão de outros.	Níveis de Abstração Casos de uso de Negócios
6	Um caso de uso tem um número excessivo de <i>Stakeholders</i> ou é confuso quanto ao entendimento do objetivo.	Múltiplos Objetivos Casos de uso de Negócios Avaliação dos Nomes
7	Uma funcionalidade do sistema possui operações de criação, leitura, atualização e remoção de informações com fluxos de ações similares.	Operações CRUD
8	Um subfluxo de ações se repete entre vários casos de uso.	Reuso via Inclusão
9	Um fluxo alternativo para um caso de uso é modelado à parte, como outro caso de uso.	Inclusão × Extensão Reuso via Inclusão
10	Um fluxo opcional de um caso de uso é incluído dentro de outro caso de uso.	Inclusão × Extensão Reuso via Inclusão
11	O modelo de casos de uso apresenta um número excessivo de casos de uso que se relacionam entre si.	Decomposição Funcional
12	Um caso de uso apresenta um subfluxo de ações muito pequeno, com uma ou duas linhas de texto.	Decomposição Funcional
13	Um modelo de casos de uso apresenta um relacionamento que indica ordem de execução entre casos de uso	Avaliação de Pré-Condições
14	A ordem de execução entre os casos de uso é complexa e difícil de ser entendida.	Outros Diagramas UML
15	O fluxo de ações de um caso de uso é muito extenso.	Casos de Uso Extensos Regras de Negócio Casos de uso de Negócios
16	O caso de uso tem detalhes da interface do usuário ou qualquer outro requisito que pode ser considerado não funcional	Aspectos não funcionais
17	A descrição de um caso de uso apresenta jargões, conceitos desconhecidos ou termos redundantes.	Glossário
18	A descrição de um caso de uso simples é confusa, pouco objetiva e apresenta erros de sintaxe.	Perfil dos Autores Múltiplos Objetivos
19	O fluxo de ações é descrito no futuro	Avaliação da Narrativa
20	Referência a pontos de Extensão	Avaliação da Narrativa
21	Referência a uma seqüência dentro do mesmo fluxo de ações	Avaliação da Narrativa
22	A estrutura da descrição dos casos de uso não segue um padrão.	<i>Templates</i>

Os detalhes de cada Tópico de revisão são apresentados no próximo item.

4.3 Desenvolvimento

Esse item apresenta o trabalho feito em cada Módulo e Tópico da Estratégia para Revisão de casos de uso com aplicações referentes ao Sistema de Gerenciamento de Processos (SGP).

4.3.1 Módulo: Estruturação

Esse módulo agrupa tópicos que, de maneira geral, agregam valor para todos os outros tópicos e somam-se ao processo de desenvolvimento. Os tópicos que compõem esse módulo, descritos abaixo, são:

- Glossário: discute a importância do uso do Glossário para a modelagem dos casos de uso;
- Perfil dos Revisores: avalia aspectos importantes na definição do profissional que trabalha com casos de uso;
- Avaliação da Narrativa: discute problemas relacionados à narração de um fluxo de ações em um caso de uso;
- *Templates*: avalia a importância em manter um formato padrão para todos os casos de uso do modelo.

4.3.1.1 Glossário

O Glossário é um artefato que tem um papel extremamente importante dentro de um processo de desenvolvimento. Segundo o UP – Processo Unificado (RATIONAL, 2003), o Glossário define termos importantes usados no projeto. Ele é um repositório central, indicado para:

- Definições de conceitos chave.
- Resolução de ambigüidade de termos e conceitos
- Explicação de jargões
- Definição de eventos de negócio
- Descrição de ações do software

Essa lista, certamente pode ser estendida com outras indicações de uso. Considerando a importância desse artefato para estabelecer um vocabulário comum, descrever casos de uso sem o auxílio do Glossário é uma tarefa que, possivelmente, irá gerar problemas de entendimento entre os *Stakeholders*.

Antes de iniciar o processo de revisão dos casos de uso do SGP, foram analisados os principais termos e conceitos que normalmente eram utilizados entre os *Stakeholders*. Como resultado dessa análise, foram identificados dezenas de termos que formalmente tinham o mesmo sentido, mas que para vários envolvidos no projeto tinham significados diferentes. Dessa forma, mesmo para aqueles casos de usos que foram modelados corretamente, existia ambigüidade de entendimento entre os *Stakeholders*.

O foco para geração do Glossário do SGP baseou-se nos seguintes conceitos:

- Objetos de negócio, representando conceitos usados no dia-a-dia da organização ou do ambiente operacional do sistema;
- Objetos do mundo real, que se relacionam com o sistema;
- Eventos gerais que acontecem no sistema.

4.3.1.2 Perfil dos Autores

Fazendo uma analogia, um bom jornalista é aquele que cria uma matéria que seja simples de ser compreendida e que não tenha dupla interpretação. Nessa comparação, a matéria do jornalista está para os leitores assim como os casos de uso estão para os *Stakeholders*.

Durante o processo de revisão dos casos de uso do SGP, as equipes foram reagrupadas seguindo o perfil dos profissionais. Os aspectos que tiveram maior relevância na avaliação dos profissionais responsáveis pela descrição dos casos de uso foram:

- Habilidade para síntese
- Abordagem sistêmica
- Razoável entendimento do domínio do processo de negócio
- Conhecimento em desenvolvimento de software e boa escrita.

A definição correta dos profissionais que revisaram o projeto do SGP foi o ponto de partida para a geração de modelos de casos de uso de alta qualidade.

4.3.1.3 Avaliação da Narrativa

Esse tópico apresenta a discussão de pequenos itens relacionados à narrativa de casos de uso, que foram mais relevantes durante a revisão do SGP.

- Narrativa no tempo presente: boa parte do fluxo de ações dos casos de uso do SGP estava descrita utilizando verbos no futuro. Essa abordagem dificulta o entendimento dos leitores dos casos de uso, principalmente para aqueles que irão dar manutenção aos modelos, depois da liberação do sistema. Descrever as ações

no presente, mesmo que o sistema ainda não exista, é mais simples e facilita a montagem de cenários dos casos de uso.

- Utilização do nome do ator nas interações: em vários casos de uso do SGP, foram encontradas descrições com referência à frase “O Ator”. Para manter clareza na leitura do caso de uso, o nome do ator deve sempre ser descrito, principalmente quando um caso de uso relaciona-se com mais de um ator.
- Especificar onde o caso de uso termina: no SGP, principalmente em casos de uso extensos, foram encontrados vários subfluxos de ações que não tinham um fim definido, gerando mal entendimento dos modelos. Dessa forma, definir onde o caso de uso termina é tão importante quanto definir como o caso de uso inicia.
- Referência a pontos de extensão: em quase todos os Casos de Uso de Extensão do SGP, foram identificadas referências a locais específicos dos Casos de Uso Base. Nessa abordagem, pequenas manutenções geraram relacionamentos de extensão inconsistentes, uma vez que trechos dos casos de uso haviam sido retirados ou movidos para outros pontos. Dessa forma, é necessário o uso dos pontos de extensão para relacionamentos de extensão, conforme exposto na seção 2.4.2 – Extensões.
- Referência lógica dentro do caso de uso: similar ao item anterior - Referência a pontos de extensão -, foram encontrados dentro do fluxo de ações de casos de uso do SGP, referência ao número do parágrafo ou linha de outra ação. Pequenas manutenções nesses modelos foram suficientes para alterar o texto e tornar inconsistentes os números dos parágrafos, o que dificultou o entendimento do fluxo. Dessa forma, é recomendável evitar referência desse tipo.
- Diagramação de fluxos complexos: de maneira geral, fluxos extensos e que apresentam muitas variações, com fluxos alternativos, são mais difíceis de serem assimilados pelos *Stakeholders*. Depois do processo de revisão dos modelos do SGP, vários diagramas UML foram criados como auxílio para leitura dos fluxos mais complexos, conforme apresentado no item 4.3.5.2 - Outros diagramas UML.
- Nível de Detalhamento: O nível de detalhamento da narrativa deve ser suficiente para a realização dos casos de uso. O grupo de padrões *Visões Ortogonais* deve ser empregado, sempre que possível, para auxiliar na composição dos diferentes níveis de abstração que um caso de uso pode apresentar.

4.3.1.4 *Templates*

Um tópico bastante importante durante o processo de revisão dos casos de uso do SGP refere-se ao uso de estruturas padrões para descrição dos casos de uso. Conforme conceituação do capítulo 2, um caso de uso é uma representação narrativa do comportamento do sistema (ARMOUR-MILLER, 2003).

Estruturar a descrição de um caso de uso em partes lógicas facilita bastante o trabalho de leitura dos *Stakeholders*. No SGP, apesar dos casos de uso apresentarem descrições com essas divisões lógicas, não existia uma padronização entre todos os casos de uso. Dessa forma, cada caso de uso parecia ter um formato próprio, dificultando ainda mais o entendimento do modelo como um todo.

Como resultado da revisão, foi adotada uma estrutura padrão para todo o projeto, (*template* de caso de uso). Existem dezenas de *templates* de casos de uso disponíveis em várias literaturas. A adoção de uma delas para o SGP foi baseada na opinião dos *Stakeholders* quanto à facilidade de entendimento dos objetivos do caso de uso. Para isso foram apresentadas três propostas de *templates* diferentes, aplicados a alguns casos de uso. O *template* escolhido seguiu o formato proposto pelo UP conforme detalhado no capítulo 2, tabela 2.1 (Estruturação de caso de uso segundo o UP).

4.3.2 Módulo: Atores

Esse módulo agrupa os tópicos que envolvem o uso de atores em contextos específicos e bastante freqüentes na modelagem de casos de uso. Os tópicos que compõe esse módulo estão descritos a seguir:

- Dispositivos como Atores: discute a criação de atores que representam dispositivos, que interagem com meio externo passivo;
- Papéis para Múltiplos Atores: discute a divisão de papéis entre os atores que interagem com um mesmo caso de uso;
- Controle de Acesso: como base da discussão, é apresentado o antipadrão que mapeia níveis de acesso como atores.

4.3.2.1 Dispositivos como Atores

O SGP possui um módulo que permite a geração de pastas, automaticamente, dentro de um fluxo, a partir da leitura de um dispositivo externo. Esse dispositivo pode ser um simples sensor de temperatura que coleta a informação em um dado momento, ou um equipamento mais sofisticado, como por exemplo, um dispositivo que monitora e armazena a quantidade de pessoas que passaram por um sensor de presença. Como definição de glossário no SGP, os dispositivos mais simples que coletam informação

do ambiente externo são chamados de Dispositivo Meio Externo. Os dispositivos mais sofisticados são considerados como sistemas externos modelados como atores. Nesse trabalho, em função da abordagem de aplicação de padrões, será apresentado apenas o modelo de Dispositivos de Meio Externo.

A Fig. 4.2 mostra o modelo para o caso de uso *Gerar Pasta via Dispositivo* antes da revisão no SGP. O primeiro problema encontrado refere-se à falta de um ator. Conforme apresentado no capítulo 2, um caso de uso nunca inicia um conjunto de ações por conta própria, conforme exposto por Kulak e Guiney (2003). O segundo problema refere-se a utilização do micro caso de uso *Ler Informação do Dispositivo*, conforme discutido na apresentação do antipadrão *Micro Casos de Uso* (item 3.1.3). Esse caso de uso caracteriza-se por apenas uma ou duas ações, responsáveis por receber a informação do dispositivo. Além disso, não existe reuso para esse pequeno fluxo de ações.

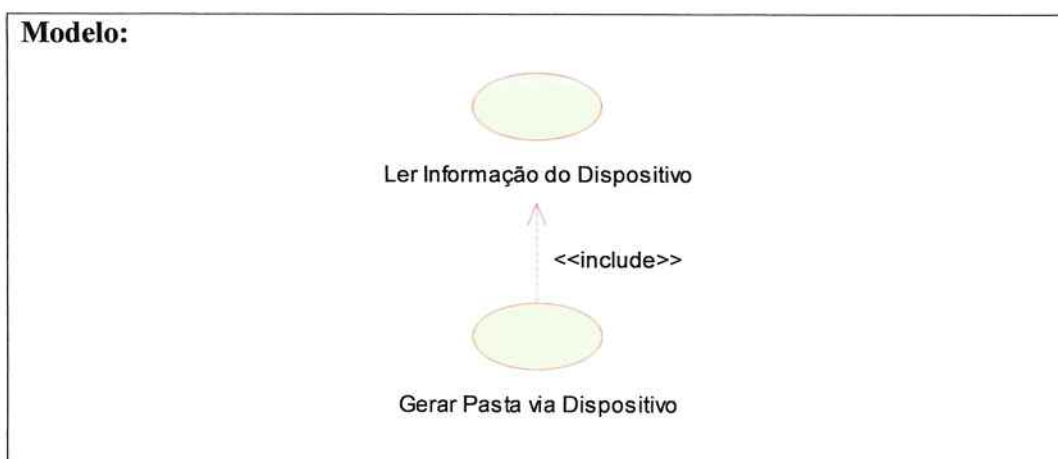


Fig. 4.2 – Caso de uso *Gerar Pasta via Dispositivo*, antes da revisão.

A Fig. 4.3 mostra o mesmo modelo revisado com a aplicação do padrão *Meio Externo Passivo*. O dispositivo é definido como um ator que passa a ter um papel ativo entre o ambiente passivo e o caso de uso *Gerar Pasta via Dispositivo*.

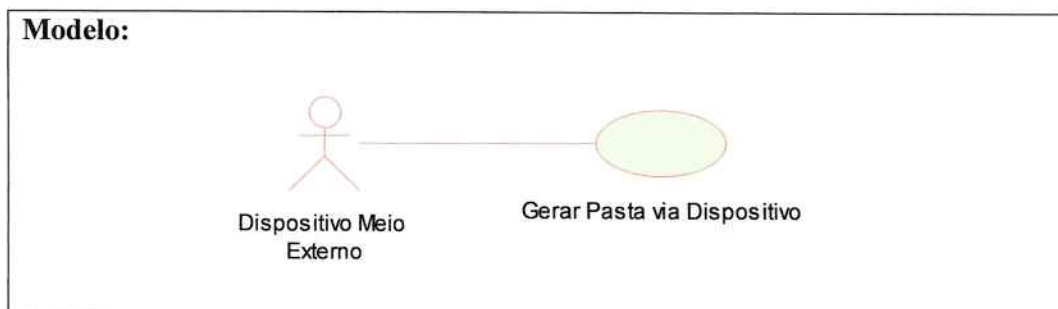


Fig. 4.3 – Caso de uso *Gerar Pasta via Dispositivo*, depois da revisão

4.3.2.2 Papéis para Múltiplos Atores

Quando se cria um modelo, é comum identificar dois ou mais atores que se relacionam com o mesmo caso de uso. Normalmente, isso significa que existem múltiplas entidades externas interessadas em utilizar o sistema conforme modelado pelo caso de uso. Entretanto, nessas situações é importante verificar se os atores representam papéis idênticos no caso de uso ou se os papéis diferem. Essa verificação é importante não somente para entender os detalhes do caso de uso, mas também para definir as interfaces do sistema. É necessário estabelecer, do ponto de vista do caso de uso, quantas instâncias externas irão interagir com cada instância do caso de uso.

O caso de uso *Redirecionar Atividade*, no SGP, permite que um participante em um passo do *Workflow* redirecione uma atividade para outro participante no mesmo passo. A Fig. 4.4 mostra o modelo e o fluxo básico antes da revisão.

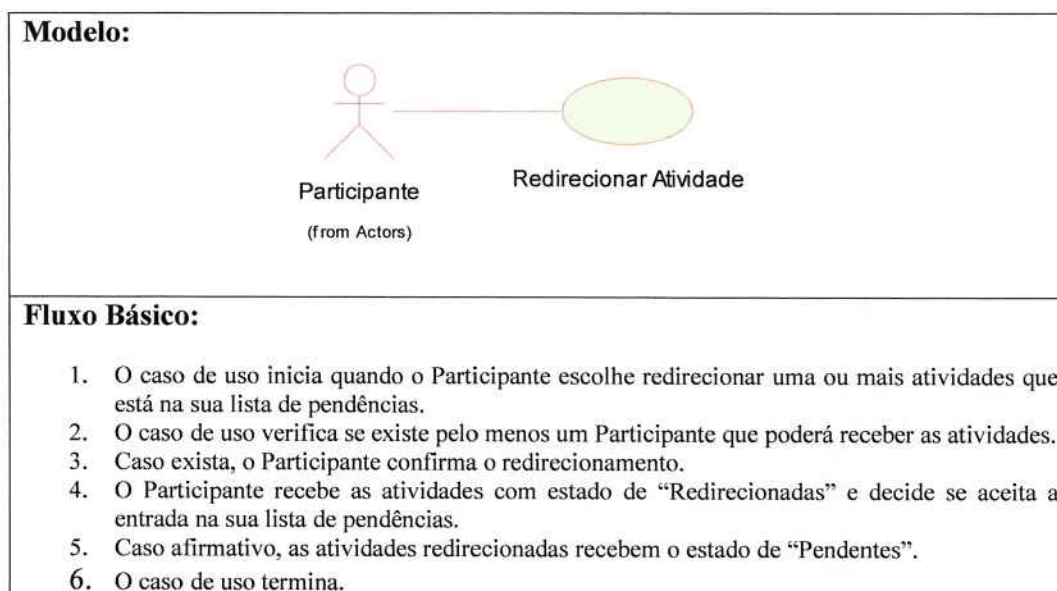


Fig. 4.4 – Modelo e fluxo básico para o caso de uso *Redirecionar Atividade*, antes da revisão

O padrão *Papéis Diferentes* sugere que se duas ou mais instâncias externas estão interagindo de maneira diferente com a instância do caso de uso, pode-se definir um ator para cada uma dessas instâncias relacionando-os com caso de uso. Dessa forma, para o caso de uso *Redirecionar Atividade*, definem-se dois atores distintos: Participante Origem e Participante Destino. A Fig. 4.5 mostra o modelo revisado através da aplicação do padrão *Papéis Diferentes*.

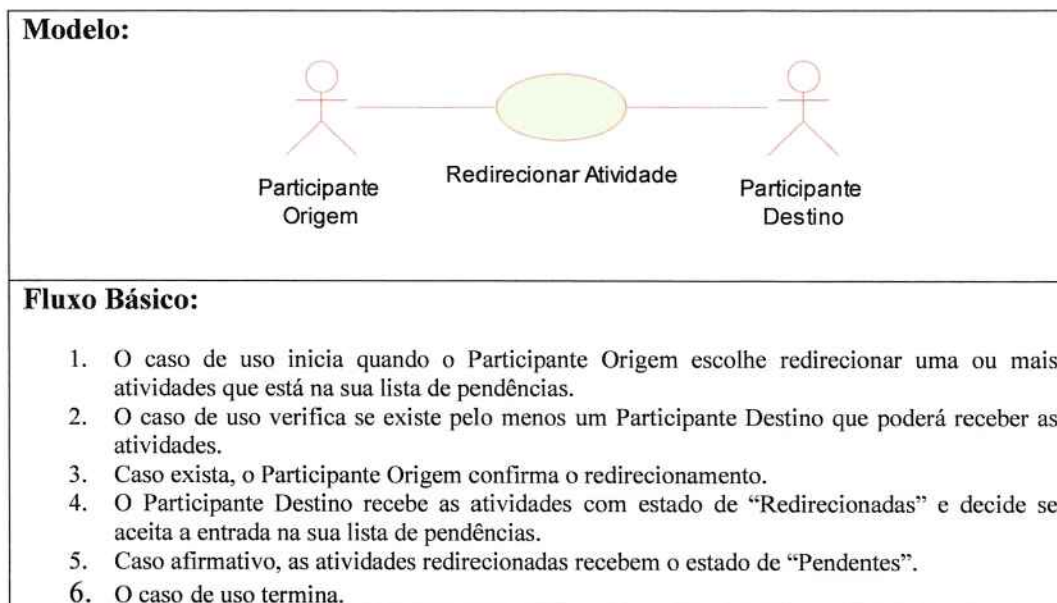


Fig. 4.5 – Modelo e fluxo básico para o caso de uso *Redirecionar Atividade*, depois da revisão

O SGP define como requisito que o Administrador e o Gestor de Fluxo podem desativar grupos que estão com estado Ativo. O caso de uso que atende esse requisito é chamado *Desativar Grupo*. A Fig. 4.6 mostra o modelo e o fluxo básico desse caso de uso antes da revisão.

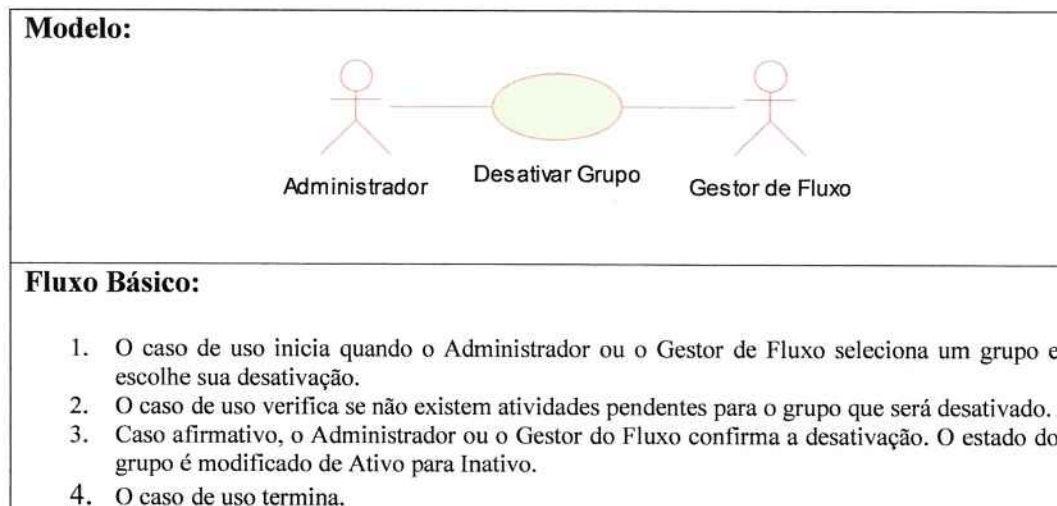


Fig. 4.6 – Modelo e fluxo básico para o caso de uso *Desativar Grupo*, antes da revisão

Diferentemente do padrão *Papéis Diferentes*, o padrão *Papel Comum* sugere que, se somente uma entidade externa está interagindo com cada instância do caso de uso, somente um ator deve estar associado com o caso de uso. Quando existem múltiplos atores associados com o mesmo caso de uso, é comum ter a impressão de que existem múltiplas entidades externas interagindo com cada instância do caso de uso. Essa impressão é incorreta e gera modelos errados. Dessa forma, para o caso de uso *Desativar Grupo*, deve-se criar um ator com papel comum chamado Gestor de Grupo. Esse novo ator deve ser associado ao caso de uso e os atores Administrador e Gestor de Fluxo devem relacionar-se a ele através de um relacionamento de herança (Generalização e Especialização). A Fig. 4.7 mostra o modelo revisado através da aplicação do padrão *Papel Comum*.

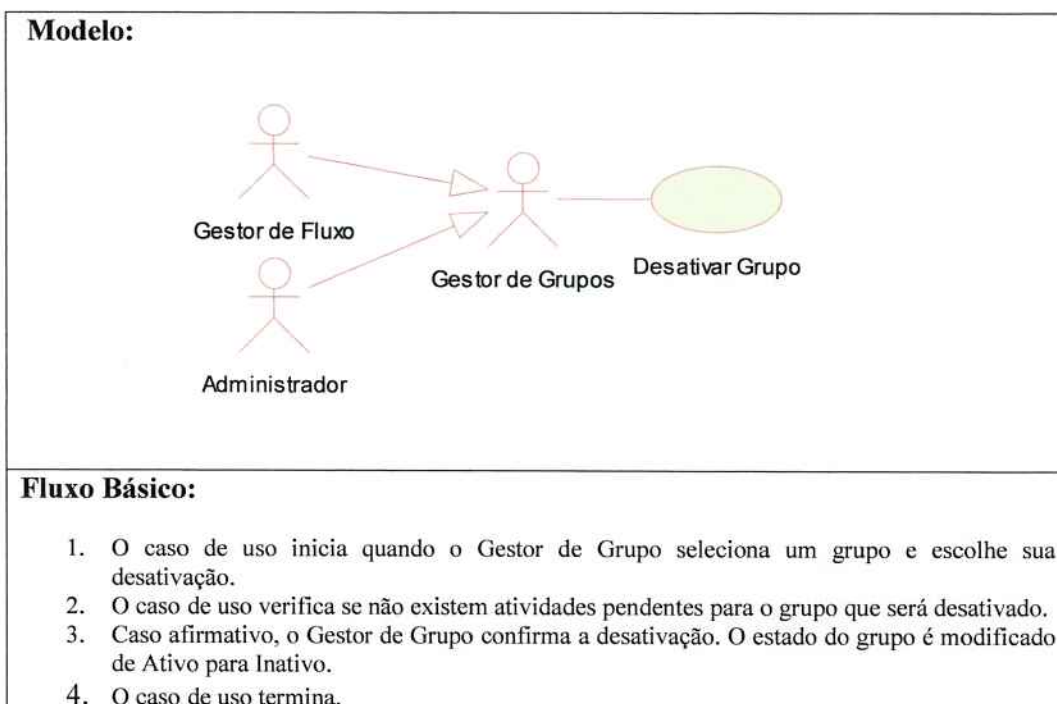


Fig. 4.7 – Modelo e fluxo básico para o caso de uso *Desativar Grupo*, depois da revisão

4.3.2.3 Controle de Acesso

No SGP, um usuário com acesso a um determinado formulário, denominado “Observador”, tem permissão apenas de leitura para os campos que compõem o formulário. Um outro usuário, “Resolvedor”, possui permissão para ler e também editar os campos do mesmo formulário.

A Fig. 4.8 mostra o modelo de casos de uso para o processo de abertura de um formulário (*Abrir Formulário para Leitura*), no SGP, antes da revisão. Esse modelo mostra um antipadrão que consiste em modelar níveis de acesso do sistema como atores.

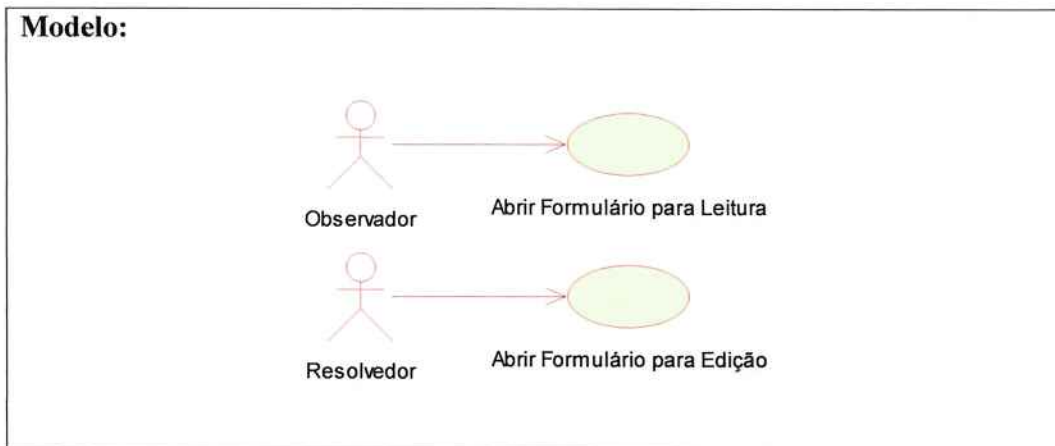


Fig. 4.8 – Modelo de casos de uso para controle de acesso a formulários no SGP, antes da revisão

Conforme conceituação no capítulo 2 e detalhamento de Antipadrões no capítulo 3, um ator é um papel representado por um usuário externo do ponto de vista do sistema. Observador e Resolvedor são níveis de acesso que um participante pode ter para acessar um determinado recurso do SGP. Dessa forma, os acessos devem ser verificados ao acessar o recurso em questão. A Fig. 4.9 mostra o modelo de casos de uso para o processo de abertura de um formulário no SGP, depois da aplicação do padrão *Checgem Explícita*.

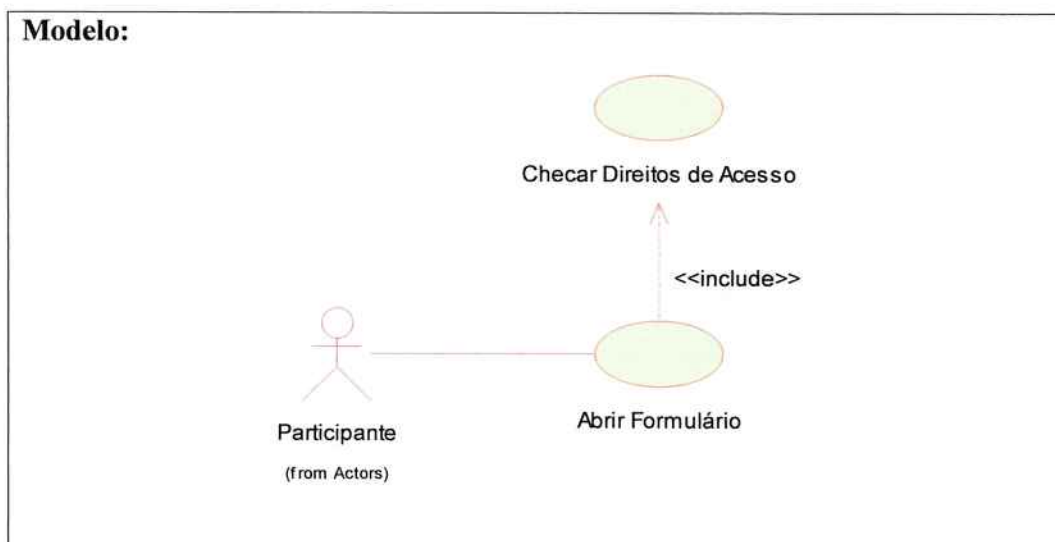


Fig. 4.9 – Modelo de casos de uso para controle de acesso de formulários no SGP, depois da revisão

4.3.3 Módulo: Objetivo e Conteúdo

Esse módulo agrupa tópicos que tem discutem uma das partes mais importantes na modelagem de casos de uso: o alinhamento entre o objetivo do caso de uso com a respectiva descrição do fluxo de ações.

- Múltiplos Objetivos: discute o uso do antipadrão *Múltiplos Objetivos* e um possível cenário para correção do problema;
- Casos de Uso de Negócio: Avalia as implicações em utilizar um antipadrão que define casos de uso com nível de abstração muito alto;
- Níveis de Abstração: como base da discussão, é apresentado o antipadrão que mistura casos de uso com níveis de detalhamento em um mesmo modelo;
- Avaliação dos Nomes: discute a formação de nomes para casos de uso

4.3.3.1 Múltiplos Objetivos

Segundo Jacobson (1999), casos de uso descrevem um conjunto seqüencial de ações, incluindo as variantes admitidas pelo sistema, para produzir um resultado observável e de valor para um ator. Na prática, um caso de uso deve ter um objetivo bastante claro. O uso do antipadrão *Múltiplos Objetivos* gera casos de uso extensos e de difícil leitura, conforme se pode ver no catálogo de antipadrões apresentado no capítulo 3, seção 3.17.

A Fig. 4.10 mostra o modelo de casos de uso para *Manutenção e Publicação de Formulário no Repositório do SGP*, antes da revisão.

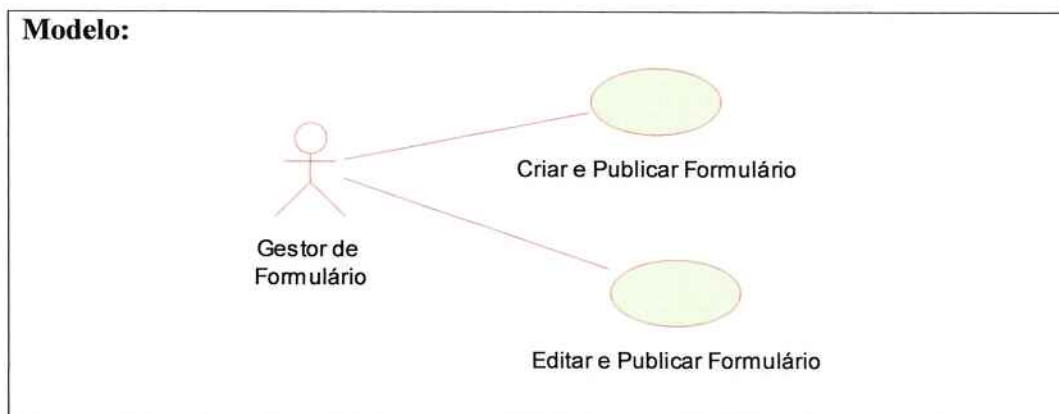


Fig. 4.10 – Modelo do caso de uso *Manutenção e Publicação de Formulário* (no Repositório do SGP), antes da revisão

“Publicar” um formulário no repositório do SGP é um processo independente da criação e edição de formulários. Dessa forma, remover o fluxo de ações relacionadas

à publicação dos casos de uso de criação e edição não traz nenhuma perda para ambos.

A Fig. 4.11 mostra o modelo de casos de uso para *Manutenção e Publicação de Formulário no Repositório do SGP*, depois da revisão. Os fluxos de criação e edição de formulário foram agrupados em apenas um caso de uso, aplicando-se o *padrão CRUD* e um novo caso de uso foi criado, denominado *Publicar Formulário*. Nesse tipo de revisão, também é bastante recomendável avaliar o uso do padrão *Atributos Comuns* para reutilização de subfluxos de ações que são comuns entre os casos de uso.

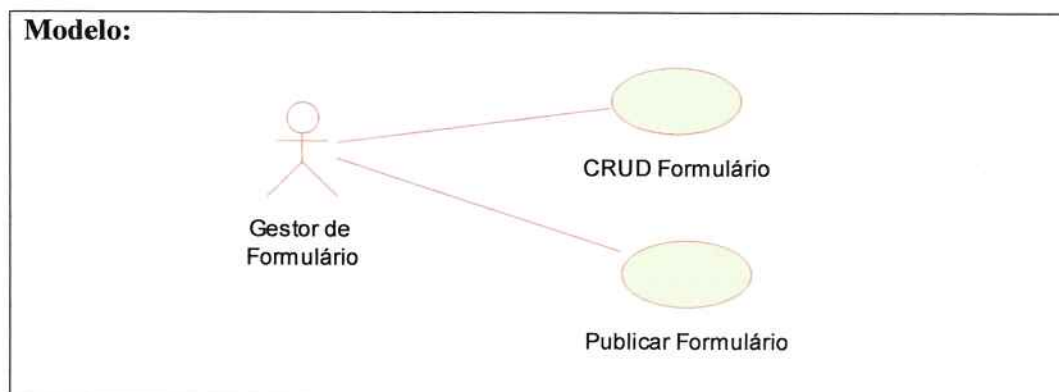


Fig. 4.11 – Modelo do caso de uso *Manutenção e Publicação de Formulário* (no Repositório do SGP), depois da revisão

4.3.3.2 Casos de Uso de Negócio

No SGP, o Gerenciamento de um Fluxo representa o processo de manutenção e acompanhamento de um fluxo após a publicação (disponibilizar o fluxo para uso dos participantes). O mapeamento desse processo para um caso de uso de sistema é errado, pois esse caso de uso não modela um uso do software, e sim um uso do negócio. O Gerenciamento do Fluxo contém vários subfluxos que fornecem valores para atores diferentes.

A Fig. 4.12 mostra o caso de uso de negócio *Gerenciar Fluxo*, usado como um caso de uso de sistema. A descrição do subfluxo de ações descreve um uso de negócio, e não de sistema.

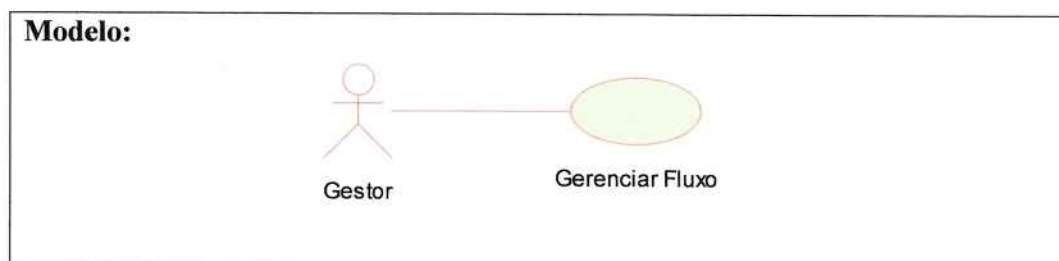


Fig. 4.12 – Modelo do caso de uso *Gerenciar Fluxo*, antes da revisão

A Fig. 4.13 mostra o modelo de gerenciamento de fluxo após a revisão. Foram extraídos os usos de sistema que fazem parte do processo de negócio *Gerenciar Fluxo*. Nesse tipo de revisão, os padrões *Casos de Usos Extensos e Seqüência* podem ser bastante úteis (Ver catálogo de padrões – seção 3.3)

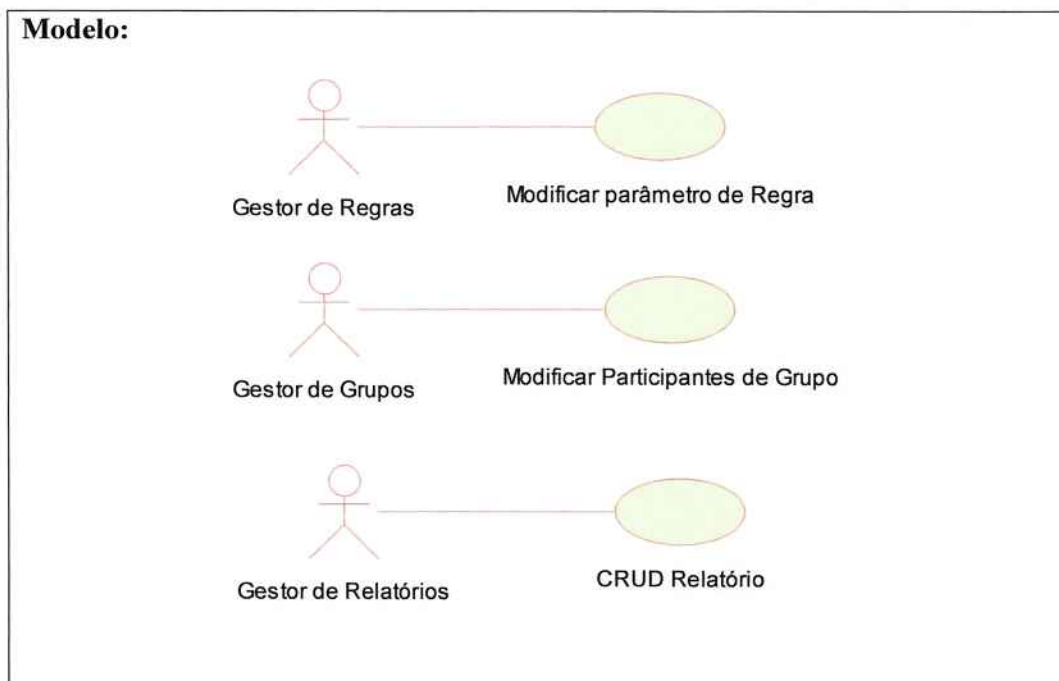


Fig. 4.13 – Modelo do caso de uso *Gerenciar Fluxo*, depois da revisão

4.3.3.3 Níveis de Abstração

Uma das maiores dificuldades durante a criação de modelos de casos de uso é estabelecer um nível de detalhamento que seja suficiente para atender aos interesses dos *Stakeholders*. Muitos modelos no SGP misturavam níveis de abstração que acabavam trazendo problemas quanto ao entendimento do modelo para alguns leitores dos casos de uso.

A Fig. 4.14 mostra o caso de uso *Exportar Fluxo* antes do processo de revisão do SGP. Esse caso de uso permite exportar todas as informações de configuração e, opcionalmente os dados que compõem as pastas de um fluxo de trabalho. O fluxo de ações do caso de uso *Verificar Disponibilidade do Servidor* é desnecessário para boa parte dos interessados porque detalha aspectos de controle da funcionalidade de exportação de fluxos. Entretanto, existem interessados como, por exemplo, os desenvolvedores, que precisam de informações sobre o cenário de uso daqueles aspectos de controle para que a funcionalidade de exportação de fluxo seja atendida por completo.

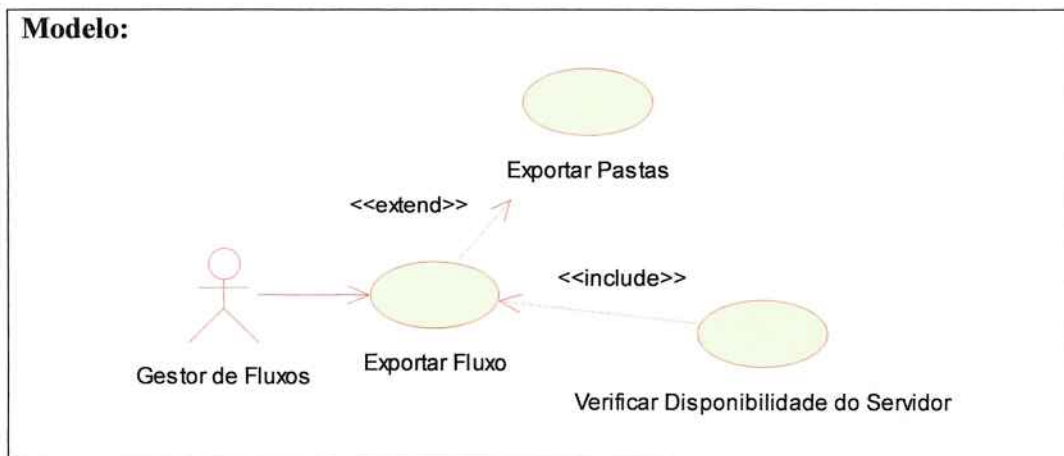


Fig. 4.14 – Modelo do caso de uso *Exportar Fluxo*, antes da revisão.

Conforme sugere o padrão *Visões Ortogonais – Descrição*, um maior detalhamento pode ser feito em uma seção separada da descrição do fluxo. O caso de uso *Verificar Disponibilidade do Servidor* foi excluído do modelo e seu fluxo de ações foi adaptado dentro de uma nova seção do caso de uso *Exportar Fluxo*. Dessa forma, o modelo final pode ser considerado dentro do mesmo nível de abstração, mantendo um entendimento para *Stakeholders* com pontos de vista diferentes. A Fig. 4.15 mostra o modelo revisado com um comentário resumindo como ficou a divisão das seções do caso de uso *Exportar Fluxo*.

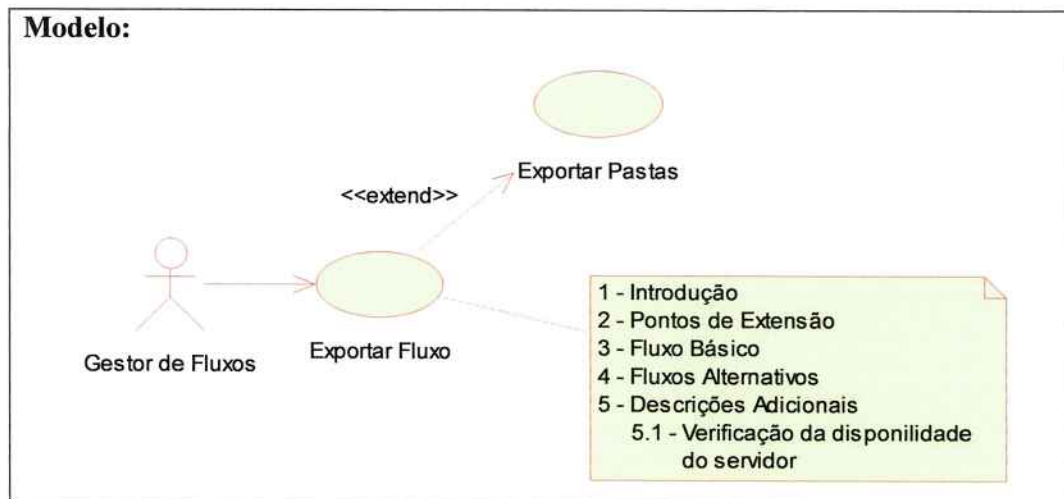


Fig. 4.15 – Modelo do caso de uso *Exportar Fluxo*, depois da revisão.

4.3.3.4 Avaliação dos Nomes

Conforme conceituação do capítulo 2, o nome de um caso de uso deve refletir o objetivo do seu conjunto de ações. No SGP foram analisados todos os nomes dos casos de uso baseado nos objetivos de cada um. A figura Fig. 4.16 mostra alguns exemplos mais relevantes de casos de uso que tiveram os nomes revisados.

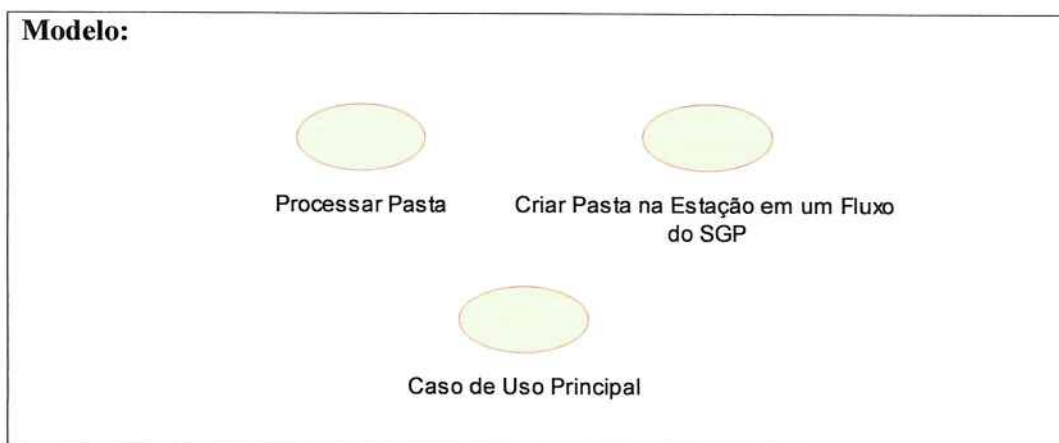


Fig. 4.16 – Exemplos de casos de uso com nomes que podem ser corrigidos

A palavra “processar” do caso de uso *Processar Pasta* dificulta o entendimento do objetivo por ter, na maioria das vezes, uma conotação genérica. No SGP, analisando o objetivo desse caso de uso, o nome mais adequado é *Enviar Pasta*.

O caso de uso *Criar Pasta na Estação em um fluxo do SGP*, está alinhado com o objetivo do fluxo de ações, entretanto, o nome é muito extenso e redundante, uma vez que toda criação de pasta, por definição, é feita em uma estação e em um fluxo. Dessa forma, o nome *Criar Pasta* já é suficiente para o alinhamento com o objetivo.

Um nome de um caso de uso pode também revelar a relevância do mesmo dentro de um modelo. A intenção do caso de uso *Caso de Uso Principal* é montar a tela principal do sistema. Nesse exemplo, ocorrem duas grandes falhas: o primeiro é associar uma característica de uma tela (tela principal) a um nome de caso de uso e o segundo erro, definir um conjunto de ações em função de uma tela do sistema. Nesse modelo, o caso de uso foi retirado do SGP.

4.3.4 Módulo: Agrupamentos e Relacionamentos

Os tópicos desse módulo mostram cenários onde casos de uso similares podem ser agrupados ou reaproveitados e algumas análises sobre o uso de relacionamentos entre casos de uso. Os tópicos que compõe esse módulo estão descritos a seguir:

- Operações CRUD: discute o uso de unificação de casos de uso para operações de criação, leitura, atualização e remoção dentro de um contexto;
- Reuso via Inclusão: avalia quando um caso de uso pode ser reaproveitado por outros casos de uso;
- Decomposição Funcional: mostra um cenário de aplicação do antipadrão homônimo do tópico e a correção a ser efetuada no modelo;
- Inclusão × Extensões: mostra um conjunto de antipadrões que são bastante comuns quando se utiliza relacionamentos de Inclusão e Extensão.

4.3.4.1 Operações CRUD

Operações para criação (*Create*), leitura (*Read*), atualização (*Update*) e remoção (*Remove*) de informações (CRUD – Iniciais em inglês) são extremamente comuns em sistemas de informação. Normalmente, são operações simples, que ao serem processadas exigem apenas uma verificação de sintaxe e tipo ou, em outros casos, algum tipo de cálculo trivial ou checagem de uma regra de negócio. A descrição do fluxo possui apenas algumas sentenças e, geralmente, não existe mais do que dois ou três caminhos alternativos no fluxo.

A aplicação do padrão CRUD Completo (item 3.3.2.1) é valiosa nesse caso, permitindo agrupar as operações e simplificar o modelo.

Muitos profissionais questionam se realmente é necessário criar e incluir casos de uso para operações do tipo CRUD na modelagem do sistema. A criação desses casos de uso é necessária, pois as operações CRUD alteram o estado do sistema e produzem um resultado observável para os atores que interagem com esses casos de uso. A não inclusão desses casos de uso gera um modelo incompleto e, provavelmente, o sistema final não atenderá alguns requisitos.

Esse agrupamento traz algumas vantagens óbvias. Primeiro, o tamanho do modelo será reduzido, pois os casos de uso serão agrupados. Segundo, raramente alguém está interessado em analisar um subconjunto dessas quatro ações como, por exemplo, “ler e apagar”, mas não em “criar e atualizar”. Agrupar esses fluxos em um simples caso de uso chamado CRUD <Nome do caso de uso> assegura que as quatro ações estejam incluídas no modelo e torna-se claro para qualquer leitor do modelo que é nesse caso de uso que estão descritas os fluxos associados a CRUD. Terceiro, o valor de cada caso de uso em separado é pequeno para os *stakeholders*, a coleção de todos eles é que traz um valor base para análise dos *stakeholders*.

No SGP, existiam vários casos de uso que seguiam o funcionamento de operações CRUD. Nesse tópico, será apresentado, como exemplo, o caso de uso para gerenciamento de *scripts*. Um *script*, no SGP, é qualquer código que pode ser executado no formulário do usuário, quando existe interação do sistema com o participante. Novos *scripts* podem ser criados, assim como alterados e excluídos. De maneira geral, essas três operações são bastante simples do ponto de vista do fluxo básico de ações. A Fig. 4.17 mostra o modelo do SGP, antes da revisão, destacando os casos de uso do CRUD. Nesse modelo, existem quatro casos de uso distintos: *Criar Script*, *Consultar Script*, *Modificar Script* e *Remover Script*, cada um deles com seu fluxo básico e fluxos alternativos. A descrição detalhada de cada um desses casos de uso não foi incluída no trabalho, entretanto, pode-se dizer que estão incluídas, praticamente na forma original, na descrição do caso de uso revisado – Fig. 4.18.

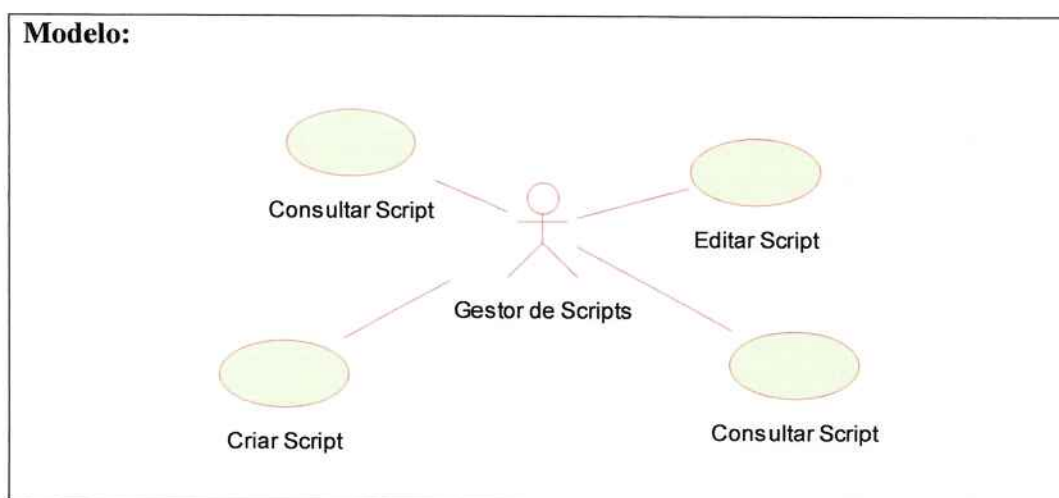
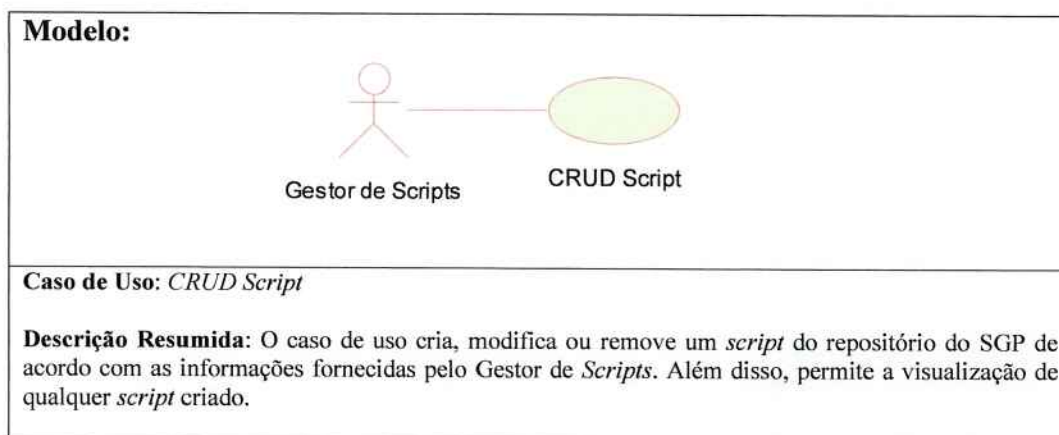


Fig. 4.17 – Modelo de Casos de Uso para gerenciamento de *scripts*, antes da revisão

A Fig. 4.18 mostra o modelo revisado, aplicando-se o padrão *CRUD – Completo*. Apenas um caso de uso e sua descrição é definido no novo modelo: *CRUD – Script*. As quatro operações básicas são descritas em quatro fluxos básicos. Tratamento de erros e exceções são tratados como fluxos alternativos do caso de uso.



Fluxo básico:

O caso de uso tem quatro diferentes fluxos básicos:

- Criar um novo *script*
- Modificar um *script* existente
- Remover um *script*
- Consultar *script*

Criar um novo script:

1. O caso de uso inicia quando o Gestor de *Scripts* escolhe criar um novo *script* para o repositório do SGP.
2. O caso de uso solicita ao Gestor de *Scripts* as seguintes informações relacionadas ao novo *script*: tipo, nome, descrição e código para execução.
3. O Gestor de *Scripts* entra com as informações requisitadas.
4. O caso de uso verifica no Repositório de *Scripts* se não existe nenhum outro com o nome informado.
5. Caso não exista, o caso de uso cria o novo *script* para o repositório e define o estado para “Ativo”.
6. O caso de uso termina.

Modificar um script existente:

1. O caso de uso inicia quando o Gestor de *Scripts* escolhe modificar um *script* do repositório do SGP.
2. O Gestor de *Scripts* modifica qualquer informação do *script*
3. O caso de uso verifica se o nome do *script* foi alterado e se não existe nenhum outro *script* no repositório com o novo nome informado.
4. Caso não exista, o caso de uso confirma as alterações para o repositório.
5. O caso de uso termina.

Remover um script:

1. O caso de uso inicia quando o Gestor de *Scripts* escolhe excluir um *script* do repositório do SGP.
2. O caso de uso busca todos os *scripts* que estão com o estado “Inativo” e apresenta para o Gestor de *Scripts*.
3. O Gestor de *Scripts* seleciona um dos *scripts*.
4. O caso de uso busca as informações relacionadas ao *script* selecionado e apresenta para o Gestor de *Scripts*.
5. Caso o Gestor de *Scripts* confirme o cancelamento, o caso de uso remove o *script*, caso contrário nenhuma alteração é realizada no Repositório de *Scripts*.
6. O caso de uso termina.

Consultar scripts:

1. O caso de uso inicia quando o Gestor de *Scripts* escolhe consultar um *script* do repositório do SGP.
2. O caso de uso solicita que seja informado um ou mais filtros para consulta: tipo, data de criação, data da última modificação e nome.
3. O Gestor de *Scripts* informa os filtros desejados.
4. O caso de uso busca todos os nomes e descrições dos *scripts* que combinam com os filtros informados
5. O caso de uso termina.

Fluxos alternativos

- em Criar um novo script - Nome já existe no repositório:

1. Se o nome do *script* já existe no repositório, o Gestor de *Scripts* é notificado e requisitado para informar um nome diferente.

Fig. 4.18 – Modelo de Caso de Uso e descrição para gerenciamento de *scripts*, depois da revisão

4.3.4.2 Reuso via inclusão

Em muitos sistemas existem subfluxos de ações que são repetidos em vários casos de uso. Conforme abordagem do capítulo 2, sobre relacionamentos entre casos de uso (item 2.4), um subfluxo comum de ações pode ser modelado, incluindo seus fluxos alternativos, em um caso de uso separado, denominado Caso de Uso Incluído (item 2.4.1). Quando esta seqüência precisar ser modificada, apenas esse caso de uso será atualizado, os demais se mantendo na forma original. O caso de uso deve conter relacionamentos de um mesmo tipo entre a modelagem do subfluxo de ações e os casos de uso em que este subfluxo aparece.

No SGP, existiam vários subfluxos que tinham o mesmo objetivo e que estavam reescritos nos diversos casos de uso. Nesse contexto, será apresentado, como exemplo, o modelo de autenticação de usuários do SGP (verificação de usuário e senha), antes e depois do processo de revisão, utilizando Padrões.

A autenticação de usuários é o ponto de partida para utilização de qualquer módulo do SGP. Ela pode ser usada também para assinar regras que serão executadas dentro dos passos de um *Workflow*. A assinatura permite que o SGP controle o acesso da regra às entidades do próprio sistema, assim como possíveis acessos a entidades externas ao SGP.

A figura Fig. 4.19 mostra o modelo antes da revisão e a descrição do caso de uso Assinar Regra.

Modelo:

Caso de Uso: *Assinar Regra*

Descrição Resumida:

O caso de uso cria uma assinatura digital para regras do SGP

Fluxo básico :

1. O caso de uso inicia quando o Gestor de Regras escolhe criar assinatura digital para regras
2. O caso de uso apresenta a lista de regras do repositório com estado “Ativo” e solicita que sejam selecionadas aquelas que serão assinadas.
3. O Gestor de Regras seleciona uma ou mais regras da lista apresentada.
4. O caso de uso monta a lista de usuários com estado “Ativo” e solicita que um usuário seja selecionado e a respectiva senha seja informada
5. O Gestor de Regras seleciona um usuário da lista apresentada.
6. O Gestor de Regras informa a senha do usuário selecionado.
7. O caso de uso carrega a senha do usuário informado, a partir do repositório de usuários do SGP
8. O caso de uso compara a senha carregada com a senha do usuário informado.
9. O caso de uso cria a assinatura digital para as regras selecionadas.
10. O caso de uso termina.

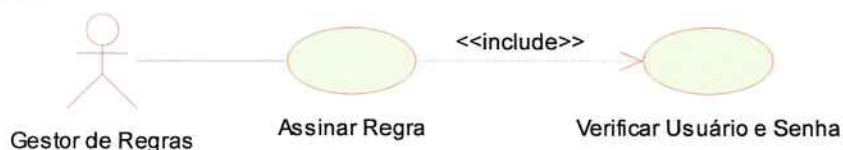
Fluxos alternativos:

- em 8 - Caso as senhas sejam diferentes, o caso de uso informa que a senha não confere e solicita novamente que a senha seja informada (passo 6).

Fig. 4.19 – Caso de uso *Assinar Regra*, antes da revisão.

Conforme sugere o padrão Reuso, se um subfluxo comum de ações é independente do contexto em que é utilizado, um relacionamento de inclusão pode ser inserido. Os Casos de Uso Base deverão ter relacionamentos de inclusão para o caso de uso que modela o subfluxo de ações. A descrição do fluxo do Caso de Uso Base deverá incorporar uma referência para o Caso de Uso Incluído onde o subfluxo é executado, mas não deverão existir referências para outras direções, pois o subfluxo comum é independente de seu contexto.

Além do caso de uso *Assinar Regra*, existem outros casos de uso que utilizam o subfluxo de verificação de usuário e senha. Dessa forma, o caso de uso *Verificar Usuário e Senha* foi criado e incluído nos casos de uso que precisam executar o mesmo subfluxo de verificação de senha e usuário. A Figura 4.20 mostra o modelo e a descrição revisada do caso de uso *Assinar Regra*.

Modelo:**Caso de Uso: *Assinar Regra***

Descrição Resumida: O caso de uso cria uma assinatura digital para regras do SGP

Ator: Gestor de Regras

Fluxo básico :

1. O caso de uso inicia quando o Gestor de Regras escolhe criar assinatura digital para regras
2. O caso de uso apresenta a lista de regras do repositório com estado “Ativo” e solicita que sejam selecionadas aquelas que serão assinadas.
3. O Gestor de Regras seleciona uma ou mais regras da lista apresentada.
4. O caso de uso monta a lista de usuários com estado “Ativo” e solicita que um usuário seja selecionado e a respectiva senha seja informada
5. O Gestor de Regras seleciona um usuário da lista apresentada.
6. O Gestor de Regras informa a senha do usuário selecionado.
7. O usuário e senha são verificados de acordo com o Caso de Uso Incluído *Verificar Usuário e Senha*.
8. O caso de uso cria a assinatura digital para as regras selecionadas.
9. O caso de uso termina.

Fluxo Alternativo:

em 7 - Se a senha não confere, o caso de uso solicita novamente que a senha seja informada.

Fig. 4.20 – Caso de uso *Assinar Regra*, depois da revisão.

4.3.4.3 Decomposição Funcional

Uma falha na modelagem de casos de uso bastante comum é o uso do Antipadrão *Decomposição Funcional*. Conforme discutido no capítulo 3 sobre esse Antipadrão, existem muitas situações onde o responsável pela criação do modelo mapeia uma função do sistema como sendo um caso de uso. Dessa forma, o modelo de casos de uso fica estruturado como se estivesse definindo um programa de computador.

A Fig. 4.21 mostra o modelo para o caso de uso *Validar Fluxo para Publicação* antes da revisão do SGP.

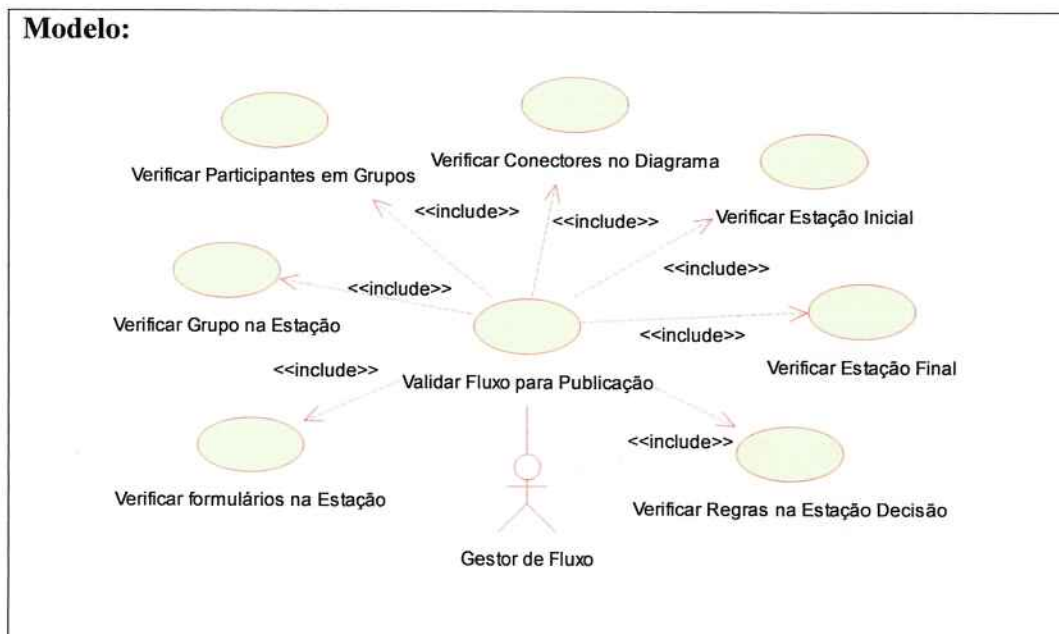
Modelo:

Fig. 4.21 – Caso de uso *Validar Fluxo para Publicação*, antes da revisão

O objetivo do caso de uso *Validar Fluxo para Publicação* é verificar se todas as regras de montagem de um fluxo foram atendidas para que o fluxo possa ser efetivamente publicado e utilizado pelos usuários do sistema. A verificação dessas regras pode ser resumida em dois subfluxos de ações (novo caso de uso): *Validar Montagem do Diagrama* e *Validar Regras de Negócio*.

A Fig. 4.22 mostra o modelo para o caso de uso *Validar Fluxo para Publicação* depois da revisão do SGP. Os dois novos casos de uso modelam usos do sistema e geram valor do ponto de vista do ator (Gestor do Fluxo). Para descrição dos novos casos de uso foram aplicados padrões dos grupo *Casos de Uso Extensos* (ver seção 3.3.5).

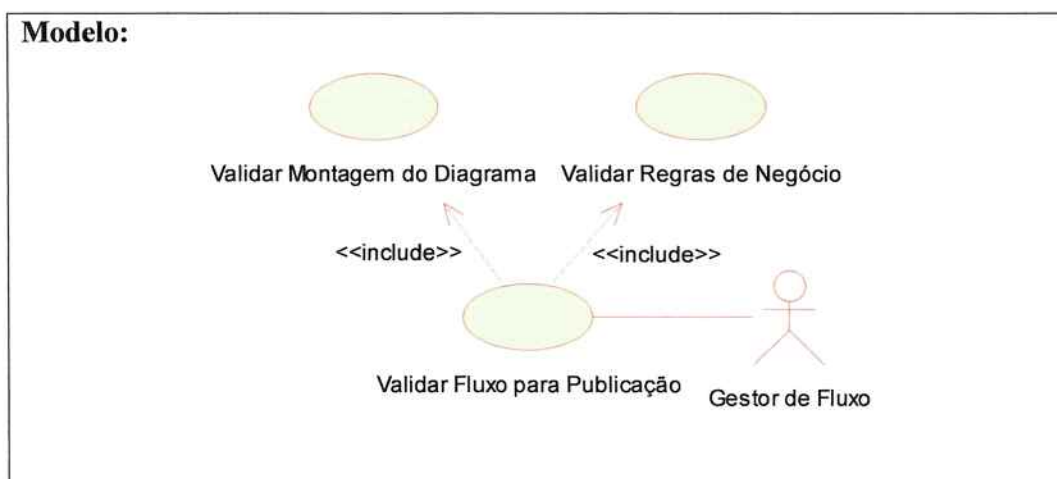


Fig. 4.22 – Caso de uso *Validar Fluxo para Publicação*, depois da revisão.

4.3.4.4 Inclusão × Extensão

Um erro comum encontrado na revisão do SGP referia-se ao uso de relacionamentos de inclusão. A Fig. 4.23 mostra o modelo e a descrição do caso de uso *Criar Grupo*. Nesse modelo, o caso de uso *Criar Participante* é associado de maneira incorreta com um relacionamento de inclusão.

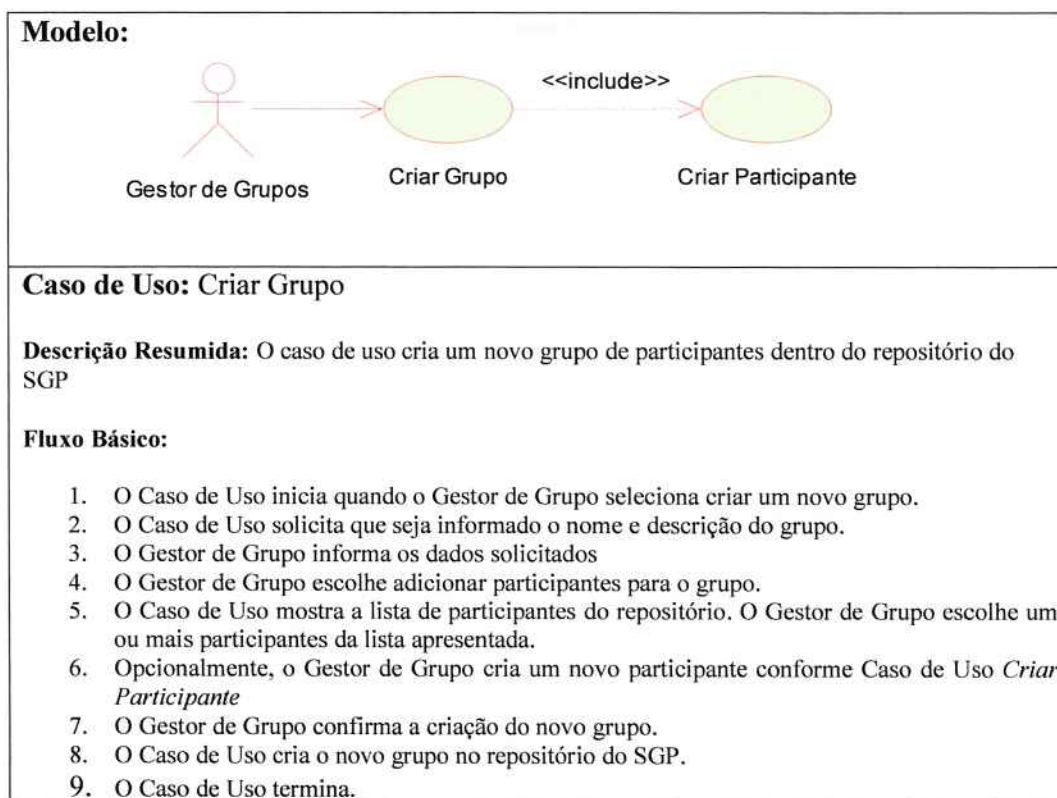
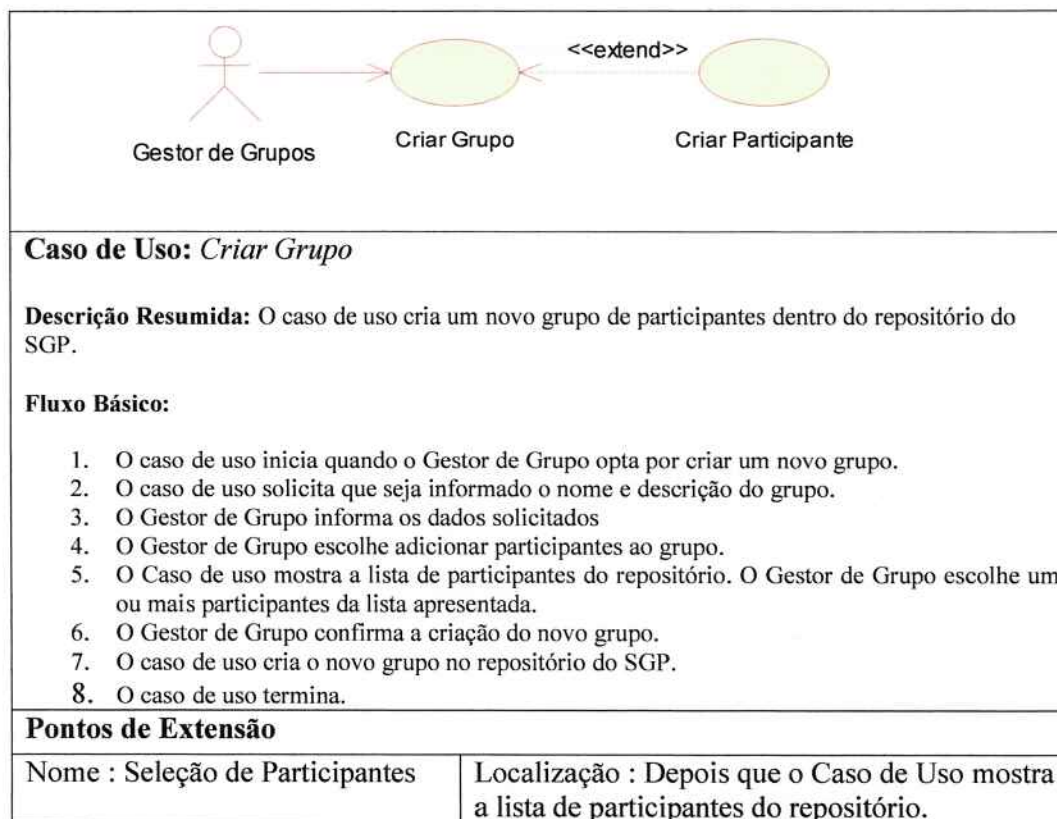


Fig. 4.23 – Caso de uso *Criar Grupo*, antes da revisão

Conforme sugere o padrão *Adição*, quando um novo subfluxo de ações é adicionada para vários casos de uso que já existem ou quando o mesmo subfluxo pode ser extraído dos casos de usos que já existem sem torná-los incompletos, um relacionamento de extensão poderá ser utilizado. Nesse contexto, casos de uso que já existem serão independentes do subfluxo comum de ações e dessa forma não conterão qualquer referência para ele. Sendo assim, o subfluxo poderá ser adicionado ou removido sem afetar as definições de qualquer caso de uso que já exista.

A Fig. 4.24 mostra o modelo e a descrição revisada do caso de uso *Criar Grupo* aplicando-se o padrão *Adição*. Nessa revisão, o caso de uso *Criar Participante* relaciona-se com o caso de uso *Criar Grupo* a partir de um relacionamento de extensão e pontos de extensão (conforme conceituação do capítulo 2). Dessa forma, o caso de uso *Criar Grupo* não gera uma referência explícita para o caso de uso *Criar Participante*. Isso garante a consistência do modelo mesmo que futuramente a criação de participantes não seja mais possível a partir da criação de um grupo. O caso de uso *Criar Participante* deverá conter uma referência ao ponto de extensão *Seleção de Participantes*.

Fig. 4.24 – Caso de uso *Criar Grupo*, depois da revisão

Conforme conceituação no capítulo 2, fluxos alternativos representam comportamentos alternativos ao fluxo básico de eventos. Geralmente, são descrições de exceções que podem acontecer no fluxo básico.

Como requisito do SGP, quando o participante informa a senha errada três vezes seguidas, o *login* desse participante deve ser bloqueado para uso no sistema. A figura Fig. 4.25 mostra o modelo do caso de uso *login*, utilizando-se o Antipadrão *Fluxo Alternativo como Extensão*.

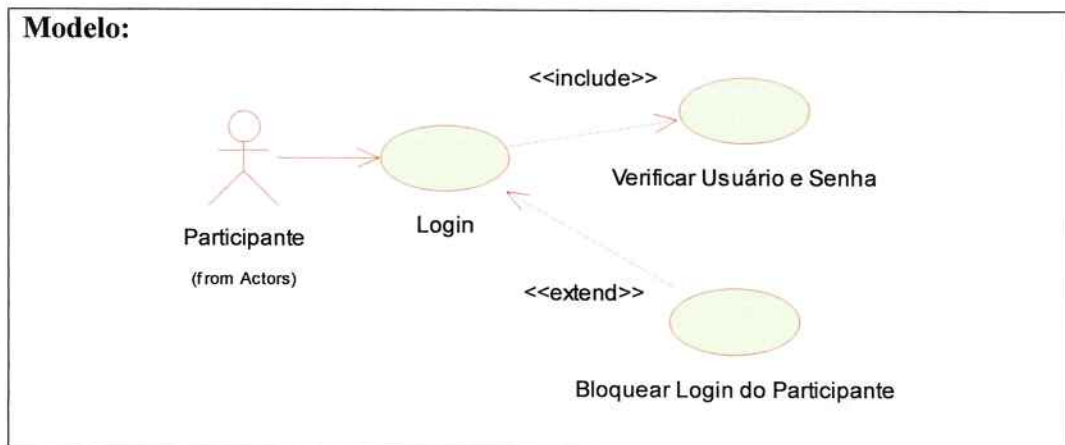


Fig. 4.25 – Caso de Uso *Login* utilizando o Antipadrão *Fluxo Alternativo como Extensão*, antes da revisão

Bloquear um *login* deve ser considerado um comportamento que acontece, excepcionalmente, quando o Participante informa por três vezes consecutivas senhas incorretas. Dessa forma, observa-se como fluxo alternativo o bloqueio do *login* do Participante. Extrair esse fluxo alternativo para dentro de um Caso de Uso de Extensão é errado, pois o comportamento extraído será executado como uma adição ao que já existe no Caso de Uso Base (*Login*) e não uma substituição a uma determinada ação.

A Fig. 4.26 mostra o modelo revisado do caso de uso *Login*. Nesse modelo, o caso de uso *Bloquear Login do Participante* é removido e sua descrição é adicionada como fluxo alternativo do caso de uso *Login*.

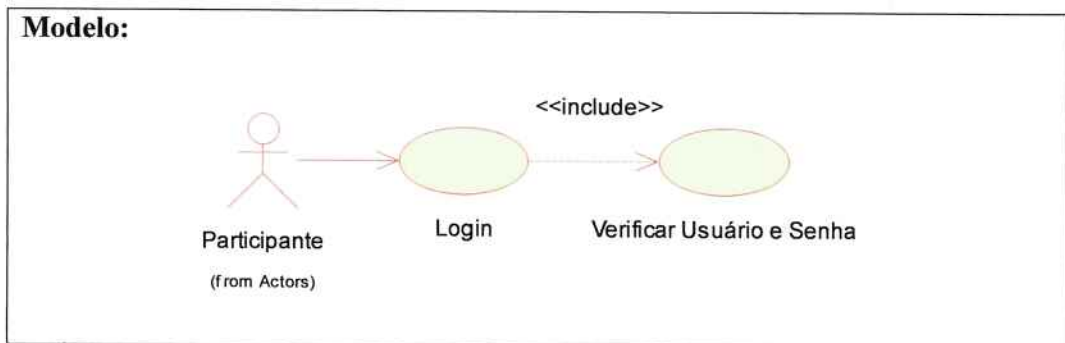


Fig. 4.26 – Caso de Uso *Login*, depois da revisão.

4.3.5 Módulo: Sequenciamento

Agrupa tópicos que envolvem a discussão sobre seqüências de execução de casos de uso. Os tópicos que compõe esse módulo estão descritos a seguir:

- Avaliação de pré-condições: discute os enganos mais comuns no uso de pré-condições;
- Outros diagramas UML: discute o uso de diagramas UML que podem auxiliar o processo de modelagem de casos de uso.

4.3.5.1 Avaliação de Pré-Condições

O engano mais comum encontrado na revisão do SGP é a utilização de relacionamentos de Inclusão e Extensão para indicar a ordem de execução dos casos de uso.

A Fig. 4.27 mostra o modelo para alguns casos de uso que utilizavam relacionamentos de Inclusão para indicar pré-condição antes do processo de revisão. Incluir o caso de uso *Login* significa incluir o sub-fluxo de ações de *Login*, ou seja, o sub-fluxo de ações passa a fazer parte do restante dos casos de uso apresentados no modelo.

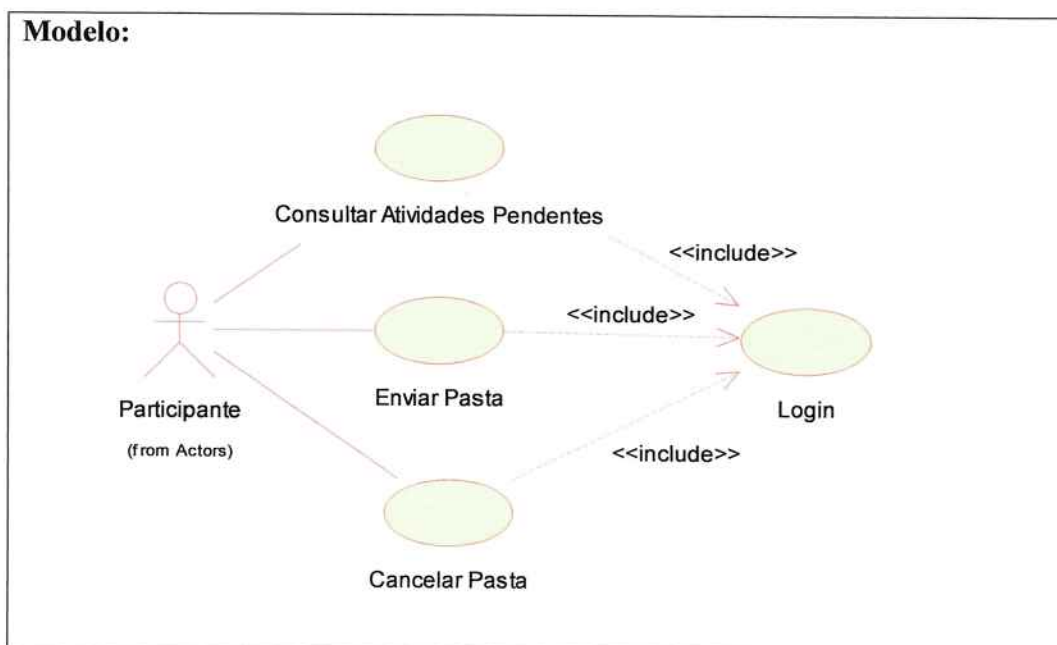


Fig. 4.27 – Modelo de casos de uso que usam relacionamentos de inclusão indicando pré-condição, antes da revisão

Conforme conceituação no capítulo 2 e definições do padrão *Sequência de Casos de uso*, a pré-condição de um caso de uso é usada para indicar em qual estado o sistema deve estar para que o caso de uso seja iniciado. Dessa forma, para esse exemplo, o *Login* já deve ter sido executado para que os outros casos de uso possam ser iniciados. A referência da pré-condição deve ser feita na própria descrição do caso de uso (conforme Tabela 2.1 - Estruturação de caso de uso segundo o UP). A Fig. 4.28 mostra o modelo revisado sem o uso de Inclusões.

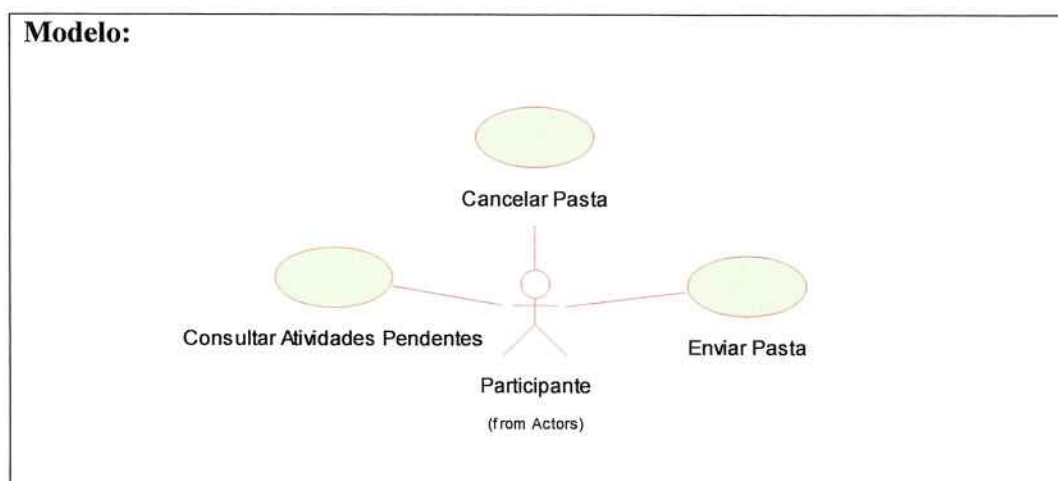


Fig. 4.28 – Modelo de casos de uso que usavam relacionamentos de inclusão indicando pré-condição, depois da revisão

4.3.5.2 Outros Diagramas UML

Além do diagrama de casos de uso, podem ser usados outros diagramas UML para facilitar o entendimento das transições dos estados do sistema, quando casos de uso são executados.

Conforme discussão no tópico *Aspectos não funcionais* do módulo de revisão *Referências*, não é recomendado utilizar detalhes de interface do usuário durante a descrição dos casos de uso. Entretanto, como os casos de uso auxiliam os projetistas de interface na montagem do projeto de usabilidade, pode-se gerar um diagrama de máquina de estados para especificar quais são os diferentes estados que a interface pode assumir e quais as transições que podem ser realizadas entre esses estados.

A Fig. 4.29 mostra uma pequena parte da máquina de estados que representa a navegação de um participante do SGP no módulo de operação. Os textos nas transições representam os nomes dos casos de uso. O ponto de partida para uso do módulo de operação é o caso de uso *Login*. Uma vez que o usuário está autenticado, pode-se escolher uma visão, através do caso de uso *Selecionar Visão*. A partir de uma visão, vários casos de uso podem ser executados e, ao final da execução, voltam para a mesma visão.

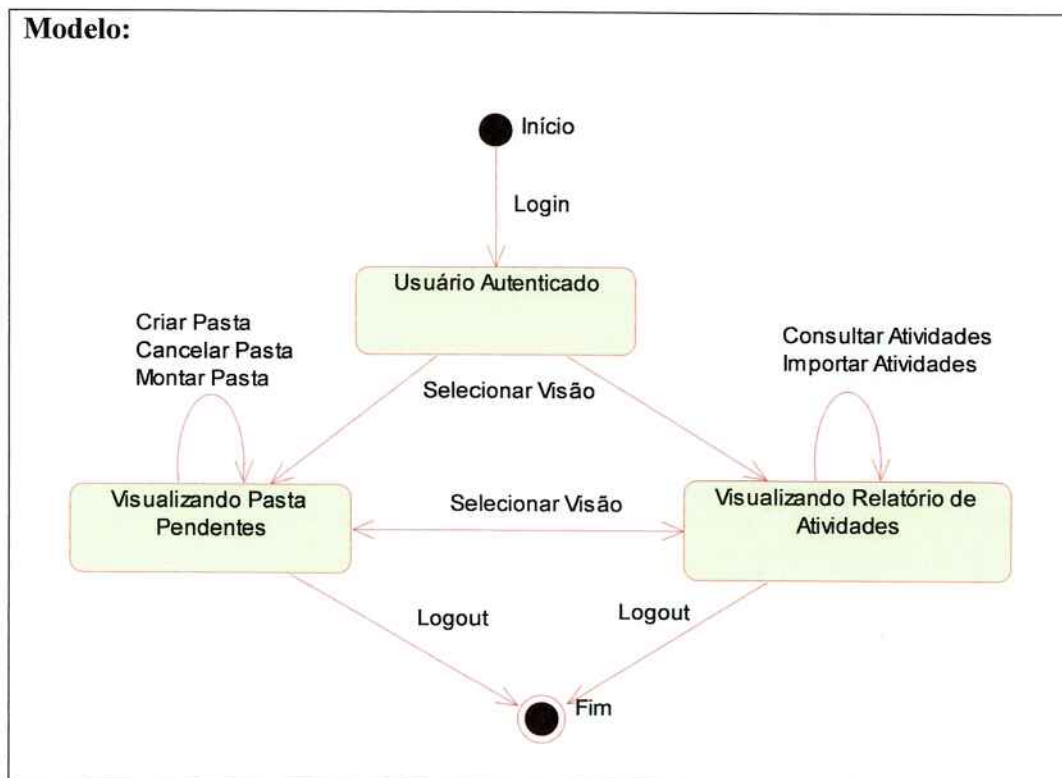


Fig. 4.29 – Máquina de estados para navegação de algumas telas do SGP

Como complemento ao Diagrama de Máquinas de Estados, foi utilizado, no SGP, o Diagrama de Atividades para facilitar o entendimento de pré-condições entre casos de uso.

4.3.6 Referências

Os tópicos desse módulo mostram cenários onde um subfluxo de ações pode ser reestruturado ou reescrito em outras partes do caso de uso para facilitar a leitura e manutenção dos modelos. Os tópicos que compõe esse módulo estão descritos a seguir:

- Caso de Uso Extensos: discute a aplicação de padrões para facilitar a leitura e o entendimento para casos de uso que são extensos;
- Regras de Negócio: discute a separação das regras de negócio do fluxo de ações dos casos de uso;
- Aspectos não funcionais: discute o uso de requisitos não funcionais nos casos de uso, como por exemplo, os da interface de usuário.

4.3.6.1 Casos de Uso Extensos

Casos de uso que possuem um fluxo de ações muito grande dificultam o entendimento do objetivo e do valor gerado para os atores associados. Conforme discussão do padrão *Casos de Usos Extensos*, não existe uma definição que ateste se um caso de uso é realmente extenso. Na prática, quanto maior o fluxo de ações, maior é a dificuldade para leitura. Dessa forma, cada caso de uso no SGP foi analisado e revisado, focando em duas variáveis: quantidade de ações e clareza quanto aos objetivos.

O exemplo escolhido para esse tópico de revisão foi o caso de uso *Criar Regra*. Considerando aspectos de definição do SGP, a criação de uma regra envolve dezenas de ações, resultando um fluxo básico de ações com várias páginas. A Fig. 4.30 mostra o modelo do caso de uso *Criar Regra* antes da revisão.

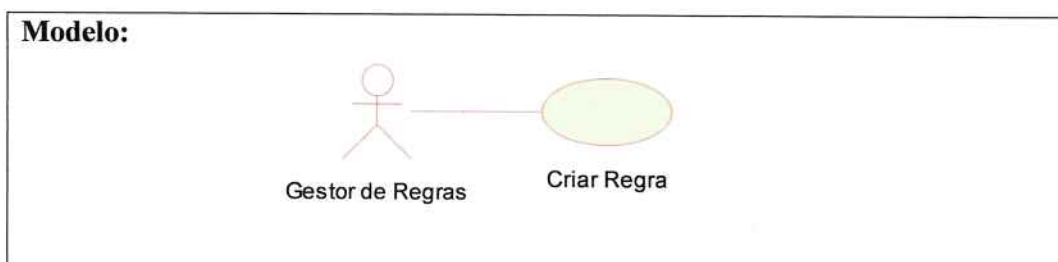


Fig. 4.30 – Caso de uso *Criar Regra*, antes da revisão.

Para aplicação do padrão *Casos de Uso Extenso: Seqüência Longa* foi identificado um conjunto de fluxos de ações que poderiam ser agrupados em seções dentro do próprio caso de uso. Algumas dessas seções, pela complexidade do fluxo, foram promovidas a novos casos de uso e referenciadas como Inclusões. As seções criadas foram: Informações básicas, Informações avançadas e Informações de Segurança. Essas seções são referenciadas a partir do fluxo básico de ações, conforme o padrão *Seqüência Longa*. As seções que foram promovidas para novos casos de uso foram: *Associar funções* e *Aprovisionar Parâmetros*. A Fig. 4.31 mostra o modelo revisado.

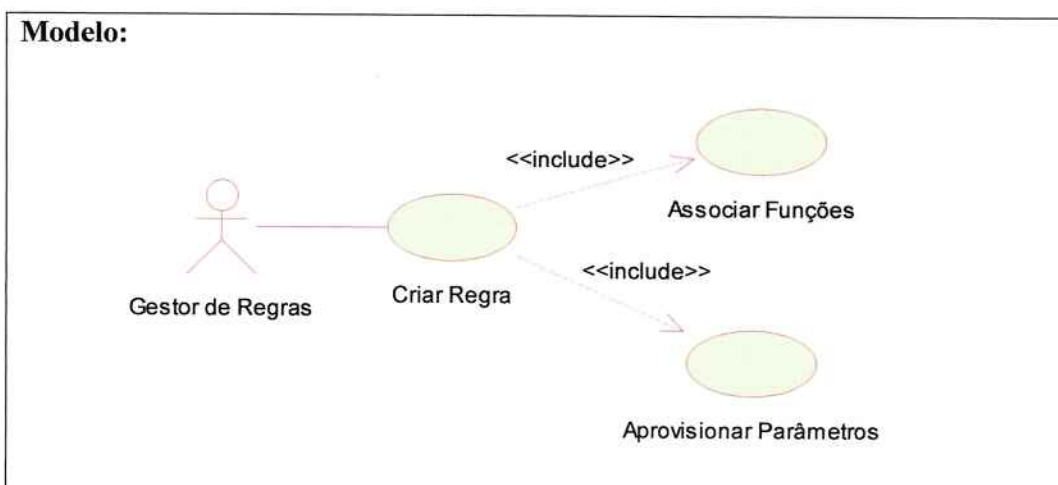


Fig. 4.31 – Modelo do caso de uso *Criar Regra*, depois da revisão

4.3.6.2 Regras de Negócio

Conforme já discutido anteriormente, o SGP é uma ferramenta para automatizar processos de negócio. As regras que fazem parte do negócio podem ser criadas dinamicamente. O modelo de casos de uso para o gerenciamento das regras é basicamente o mesmo apresentado no padrão *Regras de Negócio – Modificação Dinâmica* (seção 3.3.3.2). Nesse caso, aplica-se também o padrão *CRUD*, uma vez que além de criar novas regras, pode-se também modificar e excluir regras.

O SGP integra um pacote de sistemas de *Workflow*. Dentro desse pacote, existem outros sistemas chamados de *Business SGP* que são projetadas para uso específico por tipo de negócio. Como exemplo, existe ferramentas SGP para a área financeira e de seguros. Nessas ferramentas, as regras de negócio já fazem parte do sistema e não podem ser modificadas.

Para cada sistema *Business SGP*, foi criado um documento descrevendo as regras de negócio tendo como base, os casos de uso que estavam sendo revisados, o Glossário e o suporte dos especialistas das áreas de negócios. Nos casos de uso foi retirada qualquer informação com detalhes do negócio e, posteriormente, inseridas referências para o documento de regras de negócio. Essa separação da camada de negócio segue o padrão *Regras de Negócio – Definição Estática* (seção 3.3.3.1). O ganho na aplicação desse padrão para o SGP foi imediato: maior facilidade de leitura dos casos de uso e menor manutenção dos modelos. Além disso a separação em camadas lógicas alinhou-se com a arquitetura do sistema que também implementa as camadas separadamente.

4.3.6.3 Aspectos não Funcionais

Conforme conceituação no capítulo 2, casos de uso são a base para definição dos requisitos funcionais. Qualquer tipo de informação não-funcional detalhada no fluxo de ações pode dificultar o entendimento dos *Stakeholders* quanto aos objetivos propostos.

O erro mais comum encontrado no SGP referia-se ao uso de detalhes de interface na descrição dos casos de uso. A Fig. 4.32 mostra o fluxo básico de ações do caso de uso *Criar Pasta*, antes do processo de revisão do SGP.

Fluxo Básico:

1. O caso de uso inicia quando o participante seleciona a opção do menu principal “Criar Pasta”
2. O caso de uso apresenta a tela WEB de criação de nova pasta
3. O participante informa o assunto da pasta e seleciona no *combox* a prioridade da pasta dentro do fluxo
4. O participante pressiona o botão “OK” para confirmar a criação da nova pasta
5. O caso de uso apresenta a tela de confirmação de criação da nova pasta.
6. O caso de uso termina

Fig. 4.32 – Fluxo básico do caso de uso *Criar Pasta* com detalhamento da interface do usuário

Além de trazer informações desnecessárias do ponto de vista funcional, utilizar detalhes de interface pode amarrar o fluxo de ações com ambientes operacionais, por exemplo, Windows ou interface WEB. A Fig. 4.33 mostra o modelo revisado sem o uso de detalhes de interface.

Fluxo Básico:

1. O caso de uso inicia quando o participante seleciona criar uma nova pasta.
2. O participante informa o assunto da pasta e prioridade dentro do fluxo
3. O participante confirma a criação da nova pasta
4. O caso de uso termina

Fig. 4.33 – Fluxo básico do caso de uso *Criar Pasta* sem detalhamento de interface de usuário

Como alternativa ao uso de detalhamento de interface dentro do fluxo de ações, foi utilizada a seção Requisitos Especiais (conforme Tabela 2.1 - Estruturação de caso de uso segundo o UP). Essa seção pode referenciar qualquer documento que aborde aspectos não-funcionais como: interface de usuário, metas de desempenho, ambiente de execução, compatibilidade entre plataformas, etc.

4.3.7 Conclusões

A Estratégia para Revisão apresentada fundamenta-se no uso de padrões de casos de uso. Um padrão, por sua vez, sintetiza os conceitos e melhores práticas na construção de casos de uso. Portanto, utilizar corretamente a estratégia como uma ferramenta de apoio durante a geração dos modelos implica em ganho de produtividade e qualidade no trabalho realizado e conseqüente melhoria do produto final.

5 CONSIDERAÇÕES FINAIS

O trabalho de revisão dos casos de uso do SGP mostrou a importância em conhecer profundamente os conceitos que envolvem essa técnica de modelagem, considerando a relevância dos casos de uso para o processo de desenvolvimento.

O formato simples e a escrita em linguagem natural é um dos grandes benefícios na utilização de casos de uso, entretanto, como mostrado no estudo de caso do SGP, a aplicação inadequada dos conceitos, com o uso de padrões incorretos, resultou em um produto com uma série de problemas, principalmente defeitos oriundos da baixa qualidade da especificação.

O SGP foi desenvolvido novamente, tendo como base os modelos revisados. O produto final apresentou melhorias excepcionais de qualidade, com grande aceitação do mercado de gerenciamento de processos.

A contribuição desse trabalho foi o desenvolvimento da estratégia de revisão de casos de uso. Espera-se que a aplicação dessa estratégia, em qualquer projeto, possa gerar os mesmos benefícios que foram fundamentais na reconstrução do SGP.

Conforme abordado na introdução do capítulo 4, a Estratégia para Revisão é resultado de um trabalho para um projeto específico. Dessa forma, as Situações que compõem a estratégia foram as mais relevantes dentro dos cenários de aplicação encontrados no SGP e, certamente, se repetem em vários outros tipos de projetos, uma vez que grande parte dos sistemas apresentam características similares.

Considerando a Estratégia para Revisão como um trabalho inicial, fica como sugestão para trabalhos futuros, a adição de outros padrões, Tópicos e novas Situações, que podem ser identificadas durante o processo de revisão de casos de uso. A complementação de novos itens pode transformar a Estratégia para Revisão num verdadeiro *framework* para modelagem de casos de uso.

6 REFERÊNCIA BIBLIOGRÁFICA

CURTIS, B. The global pursuit of process maturity. IEEE Software, v. 17, n. 4, jul./ago. 2000.

LINDVALL, M. Process diversity in software development. IEEE Software, v. 17, n. 4, jul./ago. 2000.

ARMOUR, F; MILLER, G. Advanced Use Case Modeling – Software Systems. Object Technology Series. Addison-Wesley, 2003, 425p.

GAMMA, E. Design Patterns: Elements of Reusable Object-Oriented Software. Boston: Addison-Wesley, 1995.

OVERGAARD, G; PALMKVIST, K. Use Cases: Patterns and Blueprints – Software Patterns Series. Addison-Wesley. 2004. 434p.

JACOBSON, I.; GRISS, M.; JONSSON, P. Software Reuse – Architecture, Process and Organization for Business Success. New York: ACM Press. 1997. 497p.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML – Guia do Usuário. Rio de Janeiro: Campus, 2000, 472p.

WASLAWICZ, R. S. Análise e projeto de Sistemas de Informação Orientados a Objetos. Rio de Janeiro: Campus, 2004.

JACOBSON, I; The Unified Software Development Process – Object Technology Series. Addison-Wesley, 1999, 463p

BITTNER, K; SPENCE, I. Use Case Modeling - Object Technology Series. Addison-Wesley. 2002, 368p

JACOBSON, I. Object-Oriented Software Engineering : A Use Case Driven Approach, Addison-Wesley, 1992

KULAK, D; GUINEY, E. Use Cases: Requirements in Context, Second Edition, Addison-Wesley, 2003, 272p.

BOOCH G; RUMBAUGH J; JACOBSON, I. The Unified Modeling Language User Guide, Addison-Wesley, Maio/1999.

RATIONAL. RUP – Rational Unified Process, Version 2003.06.00, Rational Software Corporation, 2003

SOMMERVILLE, I. Engenharia de Software. São Paulo: Addison-Wesley, 2003. 592 p.

UML2. Unified Modeling Language: Superstructure, Version 2.0, 2004.

WfMC. The Workflow Management Coalition Specification - Terminology & Glossary, **WFMC-TC-1011**, Document Status - Issue 3.0, 1999