

# Padrões de Requisitos para Especificação de Casos de Uso em Sistemas de Informação

Gabriela T. de Souza<sup>1,2</sup>, Carlo Giovano S. Pires<sup>2</sup> e Arnaldo Dias Belchior<sup>1</sup>

<sup>1</sup>Universidade de Fortaleza  
Av. Washington Soares, 1321 – Fortaleza – CE – Brasil

<sup>2</sup>Instituto Atlântico  
Rua Chico Lemos, 946 – 60 822-780 – Fortaleza – CE – Brasil  
belchior@unifor.br, {gabi,cgiovano}@atlantico.com.br

**Abstract.** *This work presents a set of requirement patterns for information systems. These patterns are based on the use case concept and present solutions for use cases specification problems, considering maintenance (insert, update and delete), transaction and query functionalities, which are a representative part of information systems scope.*

**Resumo.** *Este trabalho apresenta um conjunto de padrões de requisitos para sistemas de informação. Esses padrões são fundamentados no conceito de casos de uso e apresentam soluções para problemas de especificação de casos de uso, considerando funcionalidades de manutenção (inclusão, alteração e exclusão), transação e consulta, que representam um volume significativo do escopo de sistemas de informação.*

## 1. Introdução

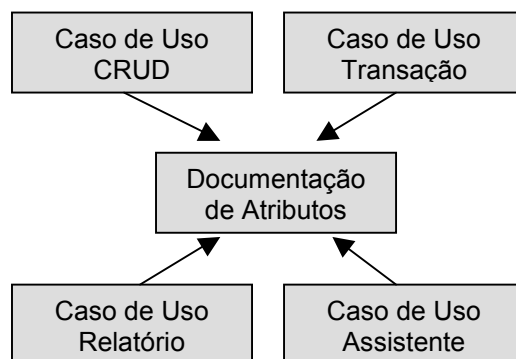
Este trabalho apresenta um conjunto de padrões de requisitos para sistemas de informação, que são fundamentados no conceito de casos de uso. Esses padrões abordam soluções para problemas de especificação casos de uso considerando questões de manutenção, consultas, relatórios e operações de transação. Isto representa um volume significativo do escopo de sistemas de informação. O relacionamento entre os padrões apresentados pode ser visto na Figura 1.

Caso de uso é um conceito amplamente difundido e utilizado para a documentação e o desenvolvimento de requisitos. Segundo o RUP [5], caso de uso é uma descrição de comportamento do sistema em termos de seqüências de ações. Um caso de uso deve produzir um resultado de valor observável para um ator. Ele contém todos os fluxos de eventos referentes à produção do "resultado de valor observável". Mais formalmente, um caso de uso define um conjunto de instâncias de casos de uso ou cenários [5]. O CMMI indica que casos de uso podem ser usados na elicitação e análise de requisitos para estabelecer os cenários operacionais do sistema [3]. Ou seja, além de representar os requisitos, os casos de uso também descrevem uma solução em alto nível.

---

<sup>1,2</sup> Copyright 2005, Gabriela T. de Souza, Carlo Giovano S. Pires e Arnaldo Dias Belchior. Permissão de cópia concedida para a Conferência Sugarloaf-PLoP 2005. Todos os outros direitos reservados.

Este trabalho utiliza um formato de caso de uso definido pela Rational [5], que compreende seções como fluxos básicos e alternativos, subfluxos de execução, requisitos especiais e regras de negócio. Serão apresentados os seguintes padrões: (i) padrão caso de uso CRUD; (ii) padrão documentação de atributos; (iii) padrão caso de uso relatório; (iv) padrão caso de uso transação e (v) padrão caso de uso assistente.



**Figura 1: Relacionamento entre os padrões apresentados**

## **2. Padrão Caso de Uso CRUD**

### **2.1. Contexto**

Este padrão é utilizado para a documentação dos requisitos de manutenção em sistemas da informação, por meio do uso de modelos e especificações de casos de uso. Os requisitos de manutenção são caracterizados por operações de Inclusão, Consulta, Alteração e Exclusão.

### **2.2. Problema**

Como documentar os requisitos funcionais de inserção, atualização, exclusão e consulta de dados por meio de especificações de casos de uso?

### **2.3. Forças**

- Todo caso de uso deve demonstrar um valor observável [5]. Em alguns casos, o usuário identifica o valor observável como a manutenção da entidade. Em outros casos, o valor observável está nas operações individuais de Inclusão, Consulta, Alteração e Exclusão.
- As operações de manutenção podem ocorrer tanto sobre entidades simples, com poucos atributos, como em entidades complexas, com vários atributos e relacionamentos.
- As operações de inclusão, alteração, remoção e consulta devem ser tratadas e seus requisitos documentados. Esses requisitos incluem validação de atributos e regras de negócio.
- Os atributos mantidos de cada entidade devem ser documentados.
- Os requisitos documentados devem ser de fácil entendimento para os usuários e para a equipe de desenvolvimento.

- Uma quantidade grande de casos de uso dificulta a gestão dos requisitos e pode indicar a existência de decomposição funcional.

## 2.4. Solução

Organizar o fluxo de eventos do caso de uso em cinco subfluxos (**Fluxo básico, Incluir, Alterar, Remover e Consultar**) como se segue:

- O **Fluxo básico** descreve a condição de início e desvia o fluxo para um dos sub-fluxos, de acordo com as operações disponíveis: Incluir, Alterar, Excluir e Consultar. Condições de início indicam os eventos que provocam a execução do caso de uso. Por exemplo, em que situação a entidade deve ser mantida, se existe alguma periodicidade requerida ou alguma questão de permissão de acesso.
- Cada subfluxo descreve o cenário operacional de uma das funcionalidades: Incluir, Alterar, Remover e Consultar.
- O subfluxo **Incluir** apresenta os atributos para a inclusão e descreve o comportamento da inclusão.
- O subfluxo **Alterar** apresenta os atributos atualizáveis, exibe seus valores e descreve o comportamento da atualização. Se os atributos atualizáveis forem os mesmos apresentados no subfluxo **Incluir**, pode-se referenciar este subfluxo.
- O subfluxo **Remover** descreve o comportamento da remoção e documenta as restrições da exclusão da entidade. Por exemplo, se alguma regra de negócio deve ser acionada ou se uma confirmação para a exclusão é exigida.
- O subfluxo **Consultar** documenta requisitos para localização da entidade, que atributos devem ser filtros para a consulta, quais são obrigatórios e quais atributos são exibidos no resultado.
- As validações de atributos e regras de negócio são documentadas em uma seção independente dos fluxos e subfluxos, ver o padrão Documentação de Atributo. A decisão sobre o momento no qual as validações e regras são executadas fará parte do projeto do caso de uso. No entanto, se esse momento já for identificado como um requisito claro da aplicação, a regra ou validação deve ser referenciada pelo subfluxo. As regras de negócio, tipicamente, representam requisitos de cálculos e tratamento de relacionamentos com outras entidades. As validações, tipicamente, documentam o tratamento para a obrigatoriedade de atributos e o tratamento de formato de atributos (datas, limites numéricos, entre outros).

### 2.4.1 Estrutura

#### Fluxo básico (*happy day*)

1. O caso de uso inicia quando o <nome do ator> necessita fazer a manutenção (inclusão, alteração, exclusão ou consulta) de uma <nome da entidade>. <descrever a condição de início do caso de uso>
2. De acordo com o tipo de manutenção desejado pelo <nome do ator>, um dos subfluxos é executado:

- a. Se o <nome do ator> deseja incluir uma nova <nome da entidade> , o sub-fluxo Incluir <nome da entidade> é executado.
- b. Se o <nome do ator> deseja alterar informações de uma <nome da entidade> já cadastrada, o sub-fluxo Alterar <nome da entidade> é executado.
- c. Se o <nome do ator> deseja excluir uma <nome da entidade> já cadastrada, o sub-fluxo Remover <nome da entidade> é executado.
- d. Se o <nome do ator> deseja consultar informações sobre uma ou mais <nome da entidade> cadastradas, o sub-fluxo Consultar <nome da entidade> é executado.

#### **Subfluxo Incluir <nome da entidade>**

1. Este subfluxo inicia quando o <nome do ator> solicita incluir um <nome da entidade>;
2. O sistema solicita ao <nome do ator> o preenchimento dos seguintes atributos:
  - <lista de atributos>.
3. O <nome do ator> preenche os atributos acima e confirma a inclusão;
4. O sistema realiza a inclusão dos dados informados pelo <nome do ator> no passo 3;
5. O sistema exibe uma mensagem informando que a inclusão da <nome da entidade> foi efetivada com sucesso;

#### **Subfluxo Alterar <nome da entidade>**

1. Este fluxo inicia quando o <nome do ator> solicita alterar um <nome da entidade>;
2. O <nome do ator> seleciona um único <nome da entidade>;
3. O sistema solicita a alteração dos seguintes atributos:
  - <lista de atributos que podem ser alterados>
4. O <nome do ator> altera os dados desejados e confirma a alteração;
5. O sistema realiza a alteração dos dados informados no passo 4;
6. O sistema exibe uma mensagem de confirmação informando que a alteração do <nome da entidade> foi efetivada com sucesso;

#### **Subfluxo Remover <nome da entidade>**

1. Este subfluxo inicia quando o <nome do ator> solicita remover um ou mais <nome da entidade>;
2. O <nome do ator> seleciona quais <nome da entidade> deseja remover e solicita a remoção;
3. O sistema solicita a confirmação para a remoção;
4. O <nome do ator> confirma a remoção;
5. O sistema remove os <nome da entidade> confirmados;
6. O sistema exibe uma mensagem informando que a remoção dos <nome da entidade> foi efetivada com sucesso;

### **Subfluxo Consultar <nome da entidade>**

1. Este fluxo inicia quando o <nome do ator> solicita consultar <nome da entidade>;
2. O sistema solicita o preenchimento dos seguintes filtros:
  - <lista de filtros>.
3. O <nome do ator> preenche os filtros e solicita a consulta;
4. O sistema apresenta as seguintes informações dos <nome da entidade> obtidos na consulta:
  - <lista de atributos>.

### **Validações e regras de negócio**

- Esta regra se aplica a todos os sub-fluxos. Atributos obrigatórios. Se algum atributo obrigatório não tiver sido preenchido, <descrever que ações o sistema deve tomar, por exemplo, “o sistema não completará a operação e notificará ao <nome do ator>, solicitando o preenchimento”>;
- Esta regra se aplica a todos os sub-fluxos. Atributos com valores não permitidos. Se algum atributo for preenchido com valor não permitido, <descrever que ações o sistema deve tomar, por exemplo, “o sistema não completará a operação e notificará ao <nome do ator>, solicitando o preenchimento”>;
- No subfluxo **Remove**, o sistema valida os <nome da entidade> selecionados de acordo com as seguintes regras:
  - o <regras de remoção>.

### **2.5. Conseqüências**

- As operações de manutenção e seus requisitos são documentadas de forma padronizada e estruturada para os diversos tipos de entidade, melhorando o entendimento do comportamento e dos requisitos, facilitando o desenvolvimento de produtos de trabalho das fases seguintes, como por exemplo, análise, projeto e casos de teste;
- As validações e regras de negócio são documentadas de maneira estruturada, evitando omissões e destacando sua importância;
- Os atributos e informações requeridos em cada operação são documentados, facilitando o entendimento da estrutura do sistema e facilitando a modelagem de dados e prototipação de telas;
- Fornece suporte ao conceito de caso de uso definido em [5]: “todo caso de uso deve demonstrar um valor observável”. A solução utiliza o conceito de sub-fluxos para agrupar em um único caso de uso as operações de Inclusão, Consulta, Alteração e Exclusão.
- Reduz o número de casos de uso do sistema por meio do agrupamento da especificação das operações de manutenção em um único caso de uso, facilitando a gestão dos requisitos.

## 2.6. Padrões relacionados

- Padrão Documentação de Atributos:
  - o Utilizado no subfluxo **Inserir** para listar os atributos da entidade; no subfluxo **Alterar**, para descrever os atributos que podem ser alterados; e no subfluxo **Consultar**, para descrever os filtros e atributos que serão exibidos no resultado da consulta.

## 3. Padrão Documentação de Atributos

### 3.1. Contexto

Em sistemas de informação, os atributos das entidades possuem diversas características como: nome, descrição, obrigatoriedade, validações, semântica, entre outras. Portanto, a documentação desses atributos deve ser elaborada de forma que essas características não sejam esquecidas.

### 3.2. Problema

Como definir e documentar de forma padronizada os diversos atributos das entidades, que são informações necessárias durante operações CRUD?

### 3.3. Forças

- Atributos podem ser de tipos primitivos, enumerados, multivalorados ou de relacionamentos. Os atributos enumerados podem assumir um valor dentro de um domínio fixo de valores. Os atributos de relacionamentos podem assumir como valor uma referência para outras entidades cadastradas no sistema. Os atributos multivalorados podem assumir um ou mais valores referentes a outras entidades cadastradas no sistema.
- Os atributos de entidades podem fazer parte de um conjunto de parâmetros ou filtros de consulta.
- Alguns atributos podem ser opcionais e outros obrigatórios. Atributos obrigatórios devem ter tratamento adequado em caso de não preenchimento na inclusão, alteração ou consulta.

### 3.4. Solução

- Documente os atributos como uma lista itemizada associada a uma operação de consulta, inclusão ou alteração. No caso da alteração, se os atributos que podem ser alterados forem os mesmos da inclusão pode-se apenas fazer uma referência aos atributos listados na inclusão.
- Uma descrição breve do atributo deve ser fornecida, quando necessário.
- Marque com um caractere especial os atributos obrigatórios (“\*”, por exemplo).
- Para atributos que indicam relacionamento, indique que é um campo de escolha fechada e indique a fonte origem dos dados de escolha. Por exemplo: Unidade federativa (campo de escolha fechada. Valores possíveis: todas as unidades federativas cadastradas no sistema).

- Para atributos enumerados, indique que é um campo de escolha fechada e indique os valores possíveis. Por exemplo: Sexo (campo de escolha fechada. Valores possíveis: feminino e masculino).
- Para atributos multivalorados, indique que é um campo de escolha múltipla e indique a fonte origem dos dados de escolha. Por exemplo: Autor do livro (campo de escolha múltipla. Valores possíveis: todos os autores cadastrados no sistema).
- Alguns atributos possuem restrição quanto aos valores aceitos. Neste caso, deve-se documentar esta restrição juntamente com o atributo. Por exemplo, o atributo temperatura corpórea do paciente só poderá assumir valores entre 35 e 42 graus.

### 3.4.1 Estrutura

- *<atributo>. <descrição do atributo>*
- *<caractere> <atributo obrigatório>*
- *<atributo> (Campo de escolha fechada. Valores possíveis: <entidade origem dos dados>). <descrição do atributo>*
- *<atributo> (Campo de escolha fechada. Valores possíveis: <valor 1>, <valor 2>, ... <valor n>). <descrição do atributo>*
- *<atributo> (Campo de escolha múltipla. Valores possíveis: <entidade origem dos dados>). <descrição do atributo>*
- *<atributo>. <descrição da validação de valores aceitos>*

### 3.5. Exemplo

Este exemplo apresenta o subfluxo Incluir da entidade cliente de uma aplicação de CallCenter.

#### Incluir Cliente

1. Este subfluxo inicia quando o Operador de Telemarketing solicita incluir um cliente;
2. O sistema solicita ao Operador de Telemarketing o preenchimento dos seguintes atributos:
  - \* Nome;
  - \* Logradouro. Descreve a rua ou a avenida em que o cliente reside;
  - \* Número;
  - \* Bairro;
  - \* Cidade;
  - \* Estado (campo de escolha fechada. Valores possíveis: todas os estados cadastrados no sistema);
  - CPF;
  - Sexo (campo de escolha fechada. Valores possíveis: feminino e masculino).

3. O Operador de Telemarketing preenche os atributos acima e confirma a inclusão;
4. O sistema realiza a inclusão dos dados informados pelo Operador de Telemarketing no passo 3;
5. O sistema exibe uma mensagem informando que a inclusão do cliente foi efetivada com sucesso;

### **3.6. Conseqüências**

- Os diversos tipos de atributos são documentados de forma simples e padronizada.
- Os atributos obrigatórios são declarados claramente, facilitando sua identificação e tratamento da implementação e testes.

## **4. Padrão Caso de Uso Relatório**

### **4.1. Contexto**

Em sistemas de informação, uma grande quantidade de dados é armazenada freqüentemente. Neste contexto, surge a necessidade de visualizar, exportar ou imprimir dados armazenados com o objetivo de conferir, analisar e tomar decisões com base nesses dados.

### **4.2. Problema**

Como documentar os requisitos de relatórios que podem incluir a necessidade de visualizar, exportar ou imprimir dados de entidades de acordo com filtros especificados, agrupamentos, totalizações e informações a serem apresentadas?

### **4.3. Forças**

- O sistema deve permitir extrair dados em diversos formatos (tela, arquivo e impressão).
- O sistema deve tratar a estrutura do relatório, como por exemplo, disposição dos campos, cabeçalho e rodapé, tamanho da fonte e orientação do papel.
- O sistema deve tratar as necessidades para exibição dos dados, como por exemplo, se os dados devem ser agrupados, se devem ser apresentadas totalizações e se existe a necessidade de algum filtro para restringir os dados que serão apresentados.

### **4.4. Solução**

- O **Fluxo básico** descreve que atributos devem ser filtros, quais são de preenchimento obrigatório e quais atributos devem ser exibidos no cabeçalho, corpo ou rodapé.
- O **Fluxo básico** descreve a condição de início. Condições de início indicam os eventos que provocam a execução do caso de uso. Por exemplo: em que situação o relatório deve ser visualizado ou impresso; ou se existe alguma periodicidade requerida.

- Os requisitos especiais são documentados em uma seção independente dos fluxos e subfluxos. Tipicamente, devem ser documentados requisitos de exportação para diversos formatos, regras das seções (regras de agrupamento, cálculo para totalização) e opções de ordenação. Um desenho esquemático do relatório e suas seções pode também ser apresentado. Descrever também o critério para filtro ou extração de dados.

#### 4.4.1 Estrutura

##### Fluxo básico (*happy day*)

1. Este fluxo inicia quando o <nome do ator> solicita gerar o relatório <nome do relatório>. <descrever a condição de início do caso de uso>;
2. O sistema solicita o preenchimento dos seguintes filtros:
  - <lista de filtros>.
3. Uma vez que o <nome do ator> forneça a informação solicitada, uma das seguintes ações é executada:
  - Se o <nome do ator> selecionar Imprimir, <descrever ação que deve ser executada>;
  - Se o <nome do ator> selecionar Visualizar, <descrever ação que deve ser executada>;
  - Se o <nome do ator> selecionar Exportar, <descrever ação que deve ser executada>;
4. O sistema apresenta o resultado na seguinte forma:
  - Cabeçalho. <descrever as informações que devem está contidas no cabeçalho>;
  - Corpo. <descrever as informações que devem estar contidas no corpo, informando lista de atributos, seções de agrupamento, e quebra de seção>;
  - Rodapé. <descrever as informações que devem estar contidas no rodapé>;
  - Totalização. <descrever que totalizações devem ser exibidas>.

##### Requisitos especiais

- Exportar para diversos formatos. <descrever para que formatos o resultado do relatório deve ser exportado, informando os requisitos necessários para a exportação de cada formato>;
- Regras das seções. <descrever quais são as regras de agrupamento de seções e as regras para o cálculo das totalizações>;
- Opções de ordenação. <listar as opções de ordenação disponíveis e descrever os requisitos para essas ordenações>.
- Regra de extração. <expressão lógica descrevendo como os atributos de filtro e outros critérios devem ser combinados para extrair os dados corretamente>
- Modelo de desenho esquemático:

|                 |          |
|-----------------|----------|
| <Logo><Sistema> | <Título> |
|-----------------|----------|

|                              |                             |
|------------------------------|-----------------------------|
| <b>&lt;Grupo1&gt;</b>        |                             |
| <Campo 1>                    | <Campo 2>                   |
| <b>&lt;Total grupo 1&gt;</b> | <b>&lt;Soma campo 2&gt;</b> |
|                              | <Página x de y>             |

#### 4.5. Exemplo

Este exemplo apresenta um relatório de cliente de uma aplicação de CallCenter. O relatório possui totalizações por bairro.

##### **Fluxo básico (*happy day*)**

5. Este fluxo inicia quando o Operador de Telemarketing solicita gerar o relatório de clientes por bairro. Este relatório deve ser executado antes da avaliação da carteira de clientes;
6. O sistema solicita o preenchimento dos seguintes filtros:
  - *Código da filial.*
7. Uma vez que o Operador de Telemarketing forneça a informação solicitada, uma das seguintes ações é executada:
  - Se o Operador de Telemarketing selecionar Imprimir, o sistema deve apresentar a janela de configuração de impressão;
  - Se o Operador de Telemarketing selecionar Visualizar, o sistema deve apresentar uma janela com a visualização do relatório;
  - Se o Operador de Telemarketing selecionar Exportar, o sistema deve solicitar o tipo de arquivo a ser exportado e gerar o arquivo solicitado conforme padrão definido nos requisitos especiais;
8. O sistema apresenta o resultado na seguinte forma:
  - Cabeçalho. Deve conter o nome do relatório, nome da empresa, nome da filial e a data em que o relatório foi executado;
  - Corpo. Os clientes devem ser agrupados por bairro e as seções devem conter quebras de página a cada bairro. Os seguintes atributos devem ser apresentados: nome do bairro, nome, telefone e data de cadastro do cliente;
  - Rodapé. Deve conter o número da página;
  - Totalização. As totalizações devem ser efetuadas por bairro, apresentando quantos clientes existem em cada bairro.

##### **Requisitos especiais**

- Exportar para diversos formatos. Os dados deste relatório devem ser exportados para o Excel, apresentado as informações em colunas;
- Opções de ordenação. O relatório deve ser ordenado por nome do bairro e posteriormente por nome do cliente.

- Regra de extração. Devem ser apresentados no relatório todos os clientes cadastrados no sistema e que são relacionados à filial selecionada no filtro. A identificação da filial encontra-se no cadastro do cliente.
- Modelo de desenho esquemático:

|  |          |                  |
|--|----------|------------------|
| <b>Empresa de Telemarketing</b><br><b>Filial Norte América</b><br><b>Relatório de clientes por bairro</b><br><b>01/01/2005</b> |          |                  |
| <b>Bairro: Varjota</b>   |          |                  |
| Nome   | Telefone | Data de Cadastro |
| Gabriela Souza   | 32678950 | 01/01/2004       |
| Carlo Pires  | 29087654 | 23/04/2004       |
| <b>Total de clientes da Varjota:</b>   |          | <b>2</b>         |
|  |          | Página 1         |

#### 4.6. Conseqüências

- As opções e formatos para extração são descritos.
- A estrutura do relatório e das seções é documentada de forma clara e estruturada.
- Os requisitos para extração da informação são documentados.

#### 4.7. Padrões relacionados

- Padrão Documentação de Atributos:
  - o Utilizado no **Fluxo básico** para listar os filtros.

### 5. Padrão Caso de Uso Transação

#### 5.1. Contexto

Este padrão é utilizado para a documentação dos requisitos de operações que são tratadas como um comando atômico que processa várias transações. Tipicamente operações *batch* e operações que requerem apenas um comando de início do caso de uso pelo usuário tendo pouca entrada de dados e iteração com o sistema.

#### 5.2. Problema

Como documentar os requisitos de operações que possuem a execução de longa duração ou que são executadas em formato de comando atômico, dando ênfase para os requisitos especiais dessas operações?

### 5.3. Forças

- Transações que ocorrem freqüentemente em sistemas de informação possuem várias características em comum e é importante que fiquem documentadas de forma uniforme para facilitar o entendimento dos casos de uso.
- O usuário necessita de informação sobre o progresso e o tempo estimado para a conclusão da operação.
- O usuário pode não ter familiaridade com a complexidade da tarefa.
- Transações complexas podem envolver algoritmos e cálculos.
- Durante a operação o usuário pode decidir interrompê-la.

### 5.4. Solução

- Os requisitos devem documentar a duração média do tempo de execução da operação.
- O **Fluxo básico** descreve que atributos devem ser fornecidos para a execução da operação, indicando quais são obrigatórios.
- O **Fluxo básico** descreve a condição de início. Condições de início indicam os eventos que provocam a execução do caso de uso. Por exemplo, em que situação o caso de uso deverá ser executado ou se existe alguma periodicidade requerida.
- O **Fluxo básico** deve indicar que existe uma opção de cancelamento que pode ser solicitada a qualquer momento.
- Os requisitos especiais descrevem como o progresso da operação será apresentado. O progresso é tipicamente o momento restante para o término, o número das unidades processadas ou a porcentagem do trabalho feita. Tipicamente deve ser fornecido para o usuário o status da execução da operação, informando se a operação ainda está sendo executada, e quanto tempo o usuário necessitará esperar.

#### 5.4.1 Estrutura

##### **Fluxo básico (*happy day*)**

1. Este fluxo inicia quando o <nome do ator> solicita executar a <nome da transação>. <descrever a condição de início do caso de uso>;
2. O sistema solicita o preenchimento dos seguintes dados:
  - <lista de atributos de parâmetro para a transação>.
3. O <nome do ator> preenche os dados solicitados no passo 2 e confirma a execução da operação;
4. O sistema executa a operação:
  - <Operações, indicações de algoritmos e de cálculos executados na operação>

##### **Requisitos especiais**

- O progresso da operação deverá ser apresentado em <descrever a unidade ou formato em que será apresentado o progresso da operação>.

### **Regras de negócio**

- Descrição de algoritmos e cálculos eventualmente utilizados na operação.

### **5.5. Conseqüências**

- O retorno sobre o status da execução da transação é fornecido;
- Os passos da transação, algoritmos e cálculos são documentados de forma clara.

### **5.6. Variantes**

- Em transações curtas, o tratamento do progresso da operação pode ser suprimido.

### **5.7. Padrões relacionados**

- Padrão Documentação de Atributos:
  - o Utilizado no **Fluxo básico** para listar os filtros.

## **6. Padrão Caso de Uso Assistente**

### **6.1. Contexto**

Este padrão é utilizado para a documentação dos requisitos de operações complexas que são executadas em diversos passos, onde decisões ou dados necessitam serem informados em cada passo através da iteração com o usuário.

Por exemplo, o caso de uso Submeter Proposta de Seguro é realizado em três passos. No passo inicial o proponente informa a cidade e o estado onde o veículo irá circular e os dados do veículo. No segundo passo, o sistema apresenta uma lista de coberturas e preços existentes de acordo com os dados informados no passo 1. O proponente seleciona as coberturas desejadas e avança para o passo seguinte. No terceiro e último passo o sistema apresenta o preço total do seguro e solicita a conclusão da operação.

### **6.2. Problema**

Como documentar os requisitos de uma operação, na qual diversas decisões devem ser tomadas antes que a operação possa ser concluída completamente?

### **6.3. Forças**

- Para concluir a operação, diversos passos precisam ser realizados.
- Um determinado passo pode necessitar ser terminado antes que o passo seguinte possa ser feito.

## 6.4. Solução

- O **Fluxo básico** descreve o objetivo da operação e quantos passos precisam ser executados.
- O **Fluxo básico** descreve a condição de início. Condições de início indicam os eventos que provocam a execução do caso de uso. Por exemplo: em que situação o caso de uso deverá ser executado ou se existe alguma periodicidade requerida.
- O **Fluxo básico** deve indicar que existe uma opção de cancelamento que pode ser solicitada a qualquer momento.
- Cada **Subfluxo Passo <n>** deve determinar se o usuário não pode começar o passo seguinte antes de terminar o atual.

### 6.4.1 Estrutura

#### Fluxo básico (*happy day*)

1. O caso de uso inicia quando o *<nome do ator>* necessita *<nome do caso de uso>*. *<descrever a condição de início do caso de uso>*;
2. O sistema informa tipicamente o objetivo da operação e quantos passos precisam ser executados;
3. O sistema solicita que o *<nome do ator>* execute o Passo 1;
4. Uma vez que o *<nome do ator>* decida executar o Passo 1, subfluxo **Passo 1** é executado;
5. O caso de uso se encerra.

#### Sub-fluxo Passo 1

1. Este subfluxo se inicia quando o *<nome do ator>* solicita *<descrever as ações que serão executadas neste passo>*;
2. O sistema solicita ao *<nome do ator>* o preenchimento dos seguintes atributos:
  - *<lista de atributos>*.
3. O *<nome do ator>* preenche os atributos do passo 2;
4. O sistema solicita que o *<nome do ator>* execute o passo n;
5. Uma vez que o *<nome do ator>* decida executar o passo *<n>*, subfluxo Passo *<n>* é executado;

#### Sub-fluxo Passo *<n>*

1. Este subfluxo se inicia quando o *<nome do ator>* solicita *<descrever as ações que serão executadas neste passo>*;
2. Para este subfluxo ser executado os subfluxos *<passo 1, passo 2, ... passo n>* devem ter sido executados; (não está previsto haver passos opcionais (ou alternativos? Essa seqüência linear parece limitar o uso do padrão...))
3. O sistema solicita ao *<nome do ator>* o preenchimento dos seguintes atributos:
  - *<lista de atributos>*.
4. O *<nome do ator>* preenche os atributos do passo 2;
5. O sistema solicita que o *<nome do ator>* execute o passo *<n+1>* ou conclua a operação;

6. Uma vez que o <nome do ator> decida executar o passo <n+1>, subfluxo **Passo <n+1>** é executado;
7. O caso de uso retorna para o passo 5 do fluxo básico.

#### **Sub-fluxo Passo <final>**

1. Este subfluxo se inicia quando o <nome do ator> solicita <descrever as ações que serão executadas neste passo>;
2. Para este subfluxo ser executado os subfluxos <passo 1, passo 2, ... passo n> devem ter sido executados;
3. O sistema solicita ao <nome do ator> o preenchimento dos seguintes atributos:
  - <lista de atributos>.
4. O <nome do ator> preenche os atributos do passo 2;
5. O sistema solicita que o <nome do ator> conclua a operação;
6. O caso de uso retorna para o passo 5 do fluxo básico.

#### **6.5. Conseqüências**

- Organiza e documenta todos os passos que devem ser realizados para concluir uma operação complexa.
- Permite que o usuário possa realizar intervenções, decisões e configurações em estágios intermediários de uma operação complexa.

#### **6.6. Padrões relacionados**

- Padrão Documentação de Atributos:
  - o Utilizado no **Fluxo básico** e subfluxos para listar os atributos.

#### **7. Usos conhecidos**

Os padrões apresentados neste artigo têm sido utilizados na fase de elicitação de requisitos em diversos sistemas, tais como um sistema Imobiliário, um sistema de Portal web para administração e publicação de informações de acervos culturais e um sistema de CallCenter. Porém, por motivos de confidencialidade, mais detalhes dos usos conhecidos não podem ser fornecidos.

Em [5], os exemplos de casos de uso CRUD seguem estrutura similar a proposta no padrão Caso de Uso CRUD.

#### **Referências**

- [1] Alur, D.; Crupi, J.; Malks, D. Core J2EE Patterns: as melhores práticas e estratégias de design, Editora Campus, 2002.
- [2] Buschmann, F.; Meunier, R.; Rohnert, H.; Sommerlad, P. *Pattern-Oriented Software Architecture*, John Wiley & Sons, 2001.
- [3] Chrissis, M. B., Konrad, M., Shrum, S. CMMI Guidelines for Process Integration and Product Improvement. Addison-Wesley, 2004.

- [4] Gamma, E.; Helm R.; Johnson, R.; and Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995.
- [5] Rational Unified Process®, Version 2002.05.00. Rational Software Corporation, 2001.
- [6] Souza, G. T. e Pires, C. G. *Uma Família de Padrões para Sistemas de Informação Baseada no Padrão MVC*. Sugarloaf-PloP, 2004.
- [7] Souza, G. T.; Pires, C.G.; e Barros, M. *Padrões MVC para Sistemas de Informação*. Sugarloaf-PloP, 2003.
- [8] Yoder, J.W.; Johnson, R.E.; Wilson, Q.D. *Connecting Business Objects to Relational Databases*. Fifth Conference on Patterns Languages of Programs, Monticello, Illinois, EUA, 1998. (Relatório técnico n° WUCS-98-25 da Universidade de Washington).