Projeto de Software

UNIDADE 2

FASE DE CONCEPÇÃO
Disciplina de Requisitos: Casos de Uso, Diagramas de Casos de Uso e Planejamento das Iterações

Autores:

Profa. Dra Rosana Terezinha Vaccare Braga
Prof. Dr. Paulo Cesar Masiero

ICMC/USP

Professor da Disciplina:
Prof. Dr. Valter Camargo
Prof. Esp. Alexandre Pedroso Fernandes

1 - Dos requisitos para os casos de uso

Para que a análise orientada a objetos tenha sucesso, produzindo modelos que representem o mundo real o mais fielmente possível, é preciso conhecer detalhadamente o comportamento do sistema alvo.

O **"documento de requisitos"**, em geral, é escrito de maneira informal e considerando os requisitos funcionais e não funcionais que o sistema deve satisfazer. No entanto, muitas vezes ele não existe, ou existe, mas não deixa claro o comportamento do sistema diante de cada evento que ocorre, nem tampouco a interação do sistema com o mundo exterior.

Neste documento consideraremos que existe um documento de requisitos ou que ele pode ser produzido por meio de uma técnica de requisitos qualquer Portanto, pressupomos que um documento de requisitos é dado como entrada para nosso processo.

Uma forma bastante intuitiva de documentar a interação entre o sistema e o mundo exterior é por meio dos chamados "**Casos de Uso**". Assim, o primeiro passo para o desenvolvimento de um sistema é obter seu "modelo de Casos de Uso", tendo como base o documento de requisitos elaborado anteriormente. Deve-se ressaltar que não é sempre que apenas o documento de requisitos é suficiente para desenvolver os casos de uso. Ele pode ser complementado com outras atividades, como entrevistas com os interessados, análise de documentos, análise de sistemas semelhantes etc.

Conforme discutido na Unidade 2, o PU (Processo Unificado) possui uma disciplina chamada "Requisitos", para cuidar da especificação dos requisitos do sistema. Essa disciplina pode ser realizada durante as quatro fases (concepção, elaboração, construção e transição), embora com menor ênfase nas fases finais. Portanto, embora os casos de uso tomem como base o documento de requisitos, pode-se continuar a especificar as necessidades do usuário durante a elaboração detalhada dos casos de uso, já que essa elaboração exige um profundo entendimento das ações a serem executadas pelo sistema diante de cada situação possível.

Um **caso de uso**, como o próprio nome sugere, representa uma possível utilização (uso) do sistema por um **ator**, que pode ser uma pessoa, dispositivo físico, mecanismo ou subsistema que interage com o **sistema** alvo, utilizando algum de seus serviços. Por exemplo, um ator pode ser um funcionário que opera um sistema bancário, um sensor que avisa o sistema sobre a ocorrência de algum evento no mundo real, ou um subsistema que utiliza os serviços de um outro subsistema (por exemplo, um subsistema de vendas de produtos utiliza os serviços (web-services) de um subsistema de autorização de crédito para poder concluir a venda por cartão de crédito). Um caso de uso narra a interação entre o sistema e os atores envolvidos, para atingir um ou mais objetivos.

Um caso de uso pode conter vários **cenários** em que o SISTEMA e os ATORES interagem, com sucesso ou fracasso, para atingir um objetivo. Cada cenário é uma seqüência específica de ações e interações entre os atores e o sistema alvo, ou um caso particular desde o início de uma interação até sua conclusão [Larman, 2004]. Por exemplo, o cenário em que o caixa efetua uma retirada de dinheiro de uma conta corrente, com sucesso, faz parte do caso de uso "Efetuar retirada", que pode ter outros cenários em que o cliente não tem saldo suficiente, ou desiste do saque antes da conclusão.

Elaborar os casos de uso a partir do documento de requisitos nem sempre é fácil. Dependendo do formato utilizado para documentar os requisitos, pode ser pouco intuitivo identificar os atores e os casos de uso com os quais interagem. Documentos de requisitos mais focados nos aspectos técnicos do sistema podem precisar de uma revisão antes de serem utilizados para derivar os casos de uso. Outra alternativa bastante viável é elaborar os casos de uso antes de produzir o documento de requisitos, ou seja, os casos de uso podem ser utilizados como uma forma de identificar os requisitos da aplicação, ou pelo menos ajudar nessa identificação. Por fim, pode-se fazer ambas as atividades em paralelo, ou seja, à medida que os requisitos são identificados, elaboram-se tanto os casos de uso quanto o documento de requisitos.

Deve-se ressaltar que o documento de requisitos é, muitas vezes, utilizado como um contrato entre desenvolvedor e cliente. Nesse caso, se, ao elaborar os casos de uso, forem descobertos problemas com o documento de requisitos (por exemplo, redundâncias, inconsistências, omissões, etc.), deve-se tomar alguma atitude para administrar a situação, provavelmente fazendo a substituição do documento por outro atualizado. Isso envolve reavaliar o contrato, e pode acarretar riscos, tais como não cumprimento de cronogramas e estimativas de custos.

1.2 – Identificação dos atores

Vários tipos de atores podem ser identificados em um sistema, de acordo com o “tipo” de interação que existe entre o ator e o sistema. Por exemplo, um Atendente que realiza o empréstimo de um livro em uma Biblioteca é considerado um ator principal, pois toda a comunicação com o sistema computacional é feita por ele. Já o Leitor que dirige-se ao balcão de empréstimo da Biblioteca para fazer o empréstimo é um ator secundário, pois interage com o sistema por intermédio do Atendente, mas não diretamente com o sistema computacional.

Ao analisar cada requisito do sistema, seja por meio da análise do documento de requisitos, ou seja, diretamente a partir da necessidade do usuário (por exemplo, durante entrevista com o cliente que deseja desenvolver o sistema), deve-se observar atentamente quem são os atores que supostamente serão responsáveis, direta ou indiretamente, pela interação com o sistema a fim de realizar uma certa tarefa e atingir um objetivo do negócio. Muitas vezes vários atores estão envolvidos em uma mesma atividade (ou caso de uso), devendo-se classificá-los em primários e secundários. Deve-se ter cuidado para escolher apenas os atores que interessam ao sistema, isto é, que estão dentro das fronteiras do sistema. Por exemplo, em um sistema bancário, quando o Cliente deposita um cheque, que o Caixa enviará à Câmara de Compensação, apenas Cliente e Caixa fazem parte dos limites do sistema. A Câmara de Compensação está fora desses limites, e, portanto não é de interesse para o sistema representá-la. Por outro lado, se o sistema faz o controle dos cheques compensados, ou se estivermos analisando um sistema de compensação, então a Câmara de Compensação poderia ser um ator relevante.

Terminada a busca por atores, produz-se uma lista de possíveis atores, que serão atribuídos aos casos de uso na próxima etapa. Entretanto, lembre-se que podem surgir novos atores durante o detalhamento dos casos de uso, como será discutido posteriormente. Por exemplo, ao analisar o subsistema de aquisição de livros no sistema de biblioteca, surgirá o ator “Bibliotecária Chefe”.

1.3 – Identificação dos Casos de Uso

Identificados os atores, deve-se analisar a sua interação com o sistema para atingir os resultados esperados. Cada requisito do sistema deve ser analisado em busca dos eventos que ocorrem no mundo real e que dão origem a uma interação entre um ator e o sistema – os casos de uso. Conforme dito anteriormente, um caso de uso representa "uma possível utilização do sistema por um ator".

Por exemplo, em um sistema de biblioteca, o fato de o leitor dirigir-se a um balcão para efetuar a retirada de um ou mais livros representa um caso de uso, que inicia no momento em que o leitor entrega ao atendente o(s) livro(s) que deseja emprestar e termina quando o atendente lhe entrega o(s) livro(s) devidamente autorizado(s) para retirada ou o avisa que não é possível emprestar. Várias ações intermediárias ocorrem durante esse caso de uso, como por exemplo, o leitor fornece sua identificação, o atendente verifica se o livro pode ser emprestado e se o leitor está apto a emprestar livros, o livro é desmagnetizado e a data de devolução é calculada de acordo com o tipo de leitor.

Cada requisito pode corresponder a um ou mais casos de uso e um caso de uso pode referir-se a um ou mais requisitos. Considere como exemplo os requisitos da Tabela 1. O caso de uso "Emprestar livro" corresponde aos requisitos R1, R2 e R3, simultaneamente. O caso de uso R3 poderá ter correspondência com outros casos de uso, como por exemplo, "Devolver livro", porque quando o leitor devolve um livro deve-se verificar se sua situação precisa ser regularizada (caso ele esteja "não apto" em razão do atraso do livro que está sendo devolvido no momento).

Para garantir que todos os requisitos foram cobertos, sugere-se a construção de uma tabela que cruze as informações do documento de requisitos com os casos de uso, como ilustrada na Tabela 2. Isso também torna mais fácil manter a consistência entre o documento de requisitos e os casos de uso, sempre que alguma alteração ocorre em um dos dois, melhorando a rastreabilidade dos requisitos. Assim, se for mudado um requisito, sabe-se facilmente quais casos de uso devem ser revisados e vice-versa.

Tabela 1 – Parte dos requisitos para um Sistema de Biblioteca

R1. Para usar os serviços de uma biblioteca, os leitores deverão estar registrados e possuir um cartão com número de identificação e foto.
R2. O sistema deve permitir que um *leitor apto* empreste um ou mais livros, por um período de tempo que varia de 1 semana a 6 meses, dependendo do tipo de leitor (1 semana para estudantes de graduação, 15 dias para estudantes de pós-graduação e 6 meses para docentes).
R3. O leitor está apto a emprestar livros se não possuir em seu poder livros com data de devolução vencida (menor do que a data atual) e desde que o número de livros emprestados não ultrapasse o número máximo permitido, que depende do tipo de leitor (6 livros para estudantes de graduação, 10 livros para estudantes de pós-graduação e 15 livros para docentes).

Tabela 2 – Requisitos X Casos de Uso

Requisitos	Casos de Uso
R1, R2, R3	**Emprestar livro** Um leitor empresta um ou mais livros da biblioteca, por um período de tempo que depende do tipo de leitor.
R1, R3, R4	**Devolver Livro** Um leitor devolve um livro que estava em seu poder, tornando-o novamente disponível para empréstimo.
. . .	

Desenvolvedores inexperientes devem estar atentos para não confundir casos de uso com eventos ou operações do sistema (operações serão tratadas em unidades posteriores). Por exemplo, “informar o código do leitor” ou “informar os livros a serem emprestados” não são casos de uso, mas operações executadas em resposta a eventos que ocorrem no sistema. Esses eventos fazem parte de um único caso de uso, “Emprestar Livros”, neste caso.

Outro ponto importante a discutir é se devemos considerar como casos de uso as diversas consultas e cadastros do sistema. Muitos requisitos do sistema podem referir-se a simples operações de inclusão, alteração ou modificação de dados cadastrais, como por exemplo, os livros da biblioteca. Seria necessário considerar então três casos de uso: “Incluir Livro”, “Alterar dados do livro” e “Excluir Livro”? E quanto às consultas, por exemplo, “Consultar Livro por Autor”, ‘Consultar Livro por Título”, etc., seriam casos de uso? Não há uma resposta única a esta pergunta. Alguns autores preferem representar essas funções como casos de uso, pois ocorrem no sistema com freqüência, enquanto outros autores as consideram simples demais para serem representadas como casos de uso. De fato, consultas, inserções, alterações e exclusões possuem lógica simples e bem conhecida de todos, de forma que não precisam de um detalhamento maior na fase de análise do sistema. Ao mesmo tempo, são importantes para ter uma noção geral do escopo e tamanho do sistema.

Portanto, neste curso enumeraremos consultas, inserções, alterações e exclusões como casos de uso do sistema, representando-as nos diagramas de casos de uso (Seção 1.4), mas não faremos o detalhamento desses casos de uso (Seção 1.5) durante a análise do sistema. Com isso, teremos um diagrama de casos de uso que reflete as funcionalidades do sistema sem, no entanto, nos preocuparmos com a lógica interna de casos de uso triviais. Essa lógica poderá ser necessária durante o projeto do sistema, portanto prorrogaremos sua definição até lá.

1.4 – Diagramas de Caso de Uso em UML

Os Diagramas de Caso de Uso da UML têm por objetivo representar os atores e sua interação com o sistema em cada caso de uso. Um ator é representado por um “homempalito” com o nome do ator ou por um retângulo com o estereótipo «ator» e o nome do ator, conforme ilustrado na Figura 1. Um estereótipo é um mecanismo que possibilita estender componentes da UML. Por meio de um estereótipo, pode-se dar maior destaque a um componente que tem semelhança com outros, mas que possui alguma(s) característica(s) que o distingue de outros elementos do mesmo tipo. Existem vários estereótipos prédefinidos em UML, como é o caso de «ator», mas o usuário pode definir seus próprios estereótipos.

Figura 1 – Representações para o ator na UML.

O homem palito é mais freqüentemente utilizado quando o ator é uma pessoa, mas nada impede utilizá-lo quando o ator é um dispositivo físico ou um subsistema (nestes casos é mais comum utilizar a segunda alternativa).

Os casos de uso são denotados por uma elipse com o nome do caso de uso dentro ou fora dela, conforme a Figura 2. A segunda alternativa é mais comum entre as ferramentas CASE, pois fica mais fácil redimensionar apenas o texto quando ele se encontra fora da elipse do que ter que redimensionar a elipse toda.

Figura 2 – Representações para o caso de uso na UML.

Em um Diagrama de Casos de Uso são mostrados os atores do sistema e os respectivos casos de uso dos quais participa, seja como ator principal ou secundário. Na Figura 3 é ilustrado um Diagrama de Casos de Uso (parcial) para um sistema de biblioteca, no qual participam três atores: o Leitor, o Atendente e a Bibliotecária.

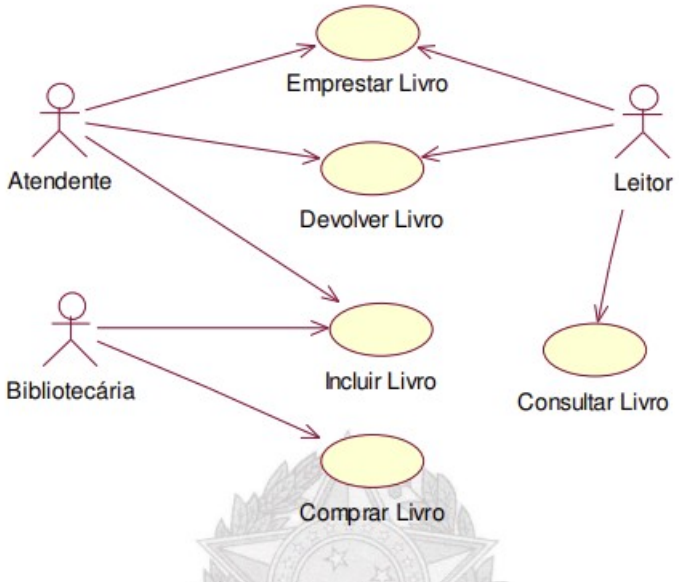


Figura 3 - Diagrama de Casos de Uso para sistema de biblioteca (parcial).

1.5 – Descrição dos Casos de uso

Enquanto os diagramas de caso de uso dão uma visão geral dos atores e de sua interação com o sistema, os casos de uso são, na verdade, descrições textuais dessa interação. O trabalho maior para elaboração dos casos de uso está justamente na escrita dos casos de uso, ao invés da construção dos diagramas em si. A descrição do caso de uso deve conter a seqüência de eventos que ocorrem tipicamente durante sua utilização, bem como as possíveis seqüências que ocorrem alternativamente, ou seja, se algumas ações podem dar erradas e não ocorrerem, ou acarretarem em outras ações diferenciadas, tudo isso deve estar documentado no caso de uso. Na Figura 4 é mostrado uma sugestão para documentar um caso de uso, proposta por Larman [2004].

Na seção “Cenário de Sucesso Principal”, deve-se descrever o caso típico em que tudo ocorre bem, isto é, trata-se do caso mais otimista, em que tudo dá certo e o objetivo do caso de uso é alcançado. Na seção “Fluxos Alternativos” analisa-se cada linha do cenário de sucesso principal, vendo o que pode dar errado e qual seria o comportamento do sistema nesses casos. Por exemplo, na biblioteca, se tudo correr bem, o leitor está apto a emprestar e o livro está disponível e sem reserva. É justamente isso que deve estar refletido na seção “Cenário de Sucesso Principal”. Depois, cada possível exceção é descrita como Fluxo alternativo, por exemplo, o leitor tem livros em atraso e por isso não pode emprestar enquanto não regularizar sua situação, ou o livro está reservado para algum outro leitor.

Quando algum passo do cenário de sucesso principal precisa ser repetido várias vezes, utiliza-se a expressão “Para cada”. Por exemplo, na Figura 4, o passo 7 é repetido para cada livro a ser emprestado pelo leitor. Pode-se também dar a noção de repetição de uma série de passos, por exemplo, um passo poderia ser redigido como “Repita os passos 7 a 9 para cada livro a ser emprestado pelo leitor”.

Caso de Uso: Emprestar Livro

Ator Principal: Atendente

Interessados e Interesses:

- Atendente: deseja registrar que um ou mais livros estão em posse de um leitor, para controlar se a devolução será feita no tempo determinado.
- Leitor: deseja emprestar um ou mais livros, de forma rápida e segura.
- Bibliotecário: deseja controlar o uso dos livros, para que não se percam e para que sempre se saiba com que leitor estão no momento.

Pré-Condições: O Atendente é identificado e autenticado.

Garantia de Sucesso (Pós-Condições): Os dados do novo empréstimo estão armazenados no Sistema. Os livros emprestados possuem status "emprestado"

Cenário de Sucesso Principal:

1. O Leitor chega ao balcão de atendimento da biblioteca e diz ao atendente que deseja emprestar um ou mais livros da biblioteca.
2. O Atendente seleciona a opção para realizar um novo empréstimo.
3. O Atendente solicita ao leitor sua carteira de identificação, seja de estudante ou professor.
4. O Atendente informa ao sistema a identificação do leitor.
5. O Sistema exibe o nome do leitor e sua situação.
6. O Atendente solicita os livros a serem emprestados.
7. Para cada um deles, informa ao sistema o código de identificação do livro.
8. O Sistema informa a data de devolução de cada livro.
9. Se necessário, o Atendente desbloqueia os livros para que possam sair da biblioteca.
10. O Leitor sai com os livros.

Fluxos Alternativos:

(1-8). A qualquer momento o Leitor informa ao Atendente que desistiu do empréstimo.

3. O Leitor informa ao Atendente que esqueceu a carteira de identificação.
   3.1. O Atendente faz uma busca pelo cadastro do Leitor e pede a ele alguma informação pessoal para garantir que ele é mesmo quem diz ser.

4. O Leitor está impedido de fazer empréstimo, por ter não estar apto.
   4.1.Cancelar a operação.

7a. O Livro não pode ser emprestado, pois está reservado para outro leitor.
   1. O Atendente informa ao Leitor que não poderá emprestar o livro e pergunta se deseja reservá-lo.
   2. Cancelar a operação (se for o único livro)

7b. O Livro não pode ser emprestado, pois é um livro reservado somente para consulta.
   1. Cancelar a operação (se for o único livro)

Figura 4 - Caso de Uso Completo para "*Emprestar Livro*"

1.6 – Formatos de casos de uso: resumido X completo

Pode-se descrever um caso de uso com maior ou menor nível de detalhes, dependendo de sua complexidade e importância de seu detalhamento para a fase corrente do PU. Por exemplo, numa fase inicial de levantamento das funcionalidades do sistema, com objetivo de fazer estimativa de tempo e custo do projeto, pode-se utilizar o formato RESUMIDO, que consiste basicamente do nome do caso de uso e um resumo sucinto, de um parágrafo, descrevendo o caso otimista (cenário de sucesso principal), como ilustrado na Figura 5. Por outro lado, na fase de elaboração do PU, o caso de uso precisa ser descrito no formato COMPLETO, para que os modelos produzidos com base neles reflitam o mais fielmente possível as funcionalidades desejadas.

Caso de uso: "Emprestar Livro"
Visão Geral: O Atendente da biblioteca realiza o empréstimo de um ou mais livros a um leitor apto a emprestar livros. O empréstimo é válido por um determinado período de tempo, de acordo com o tipo de leitor. Os livros são levados pelo leitor, depois de devidamente desmagnetizados, e marcados como "emprestados".

Figura 5 - Caso de Uso Resumido para "*Emprestar Livro*".

Pode-se diferenciar os casos de uso completos em dois casos distintos: ABSTRATO e CONCRETO. Um caso de uso no formato abstrato completo é descrito de forma a retratar todas as funcionalidades desejadas, mas sem envolver na descrição aspectos técnicos dependentes de implementação. Por exemplo, o caso de uso da Figura 4 é abstrato completo, pois nada é dito a respeito da tecnologia envolvida na identificação do leitor ou do livro.

Por outro lado, um caso de uso no formato concreto completo descreve detalhes de implementação e técnicas de software/hardware utilizadas. No exemplo do empréstimo de livro, poderia ser fornecido um projeto da interface gráfica com o usuário (GUI), com detalhes de sobre como a leitura da identificação do leitor é feita (por exemplo, por código de barras), além da indicação de como os dados são informados em cada um dos campos da GUI. Na Figura 6 é ilustrado o mesmo caso de uso, "Emprestar Livro", na versão concreta.

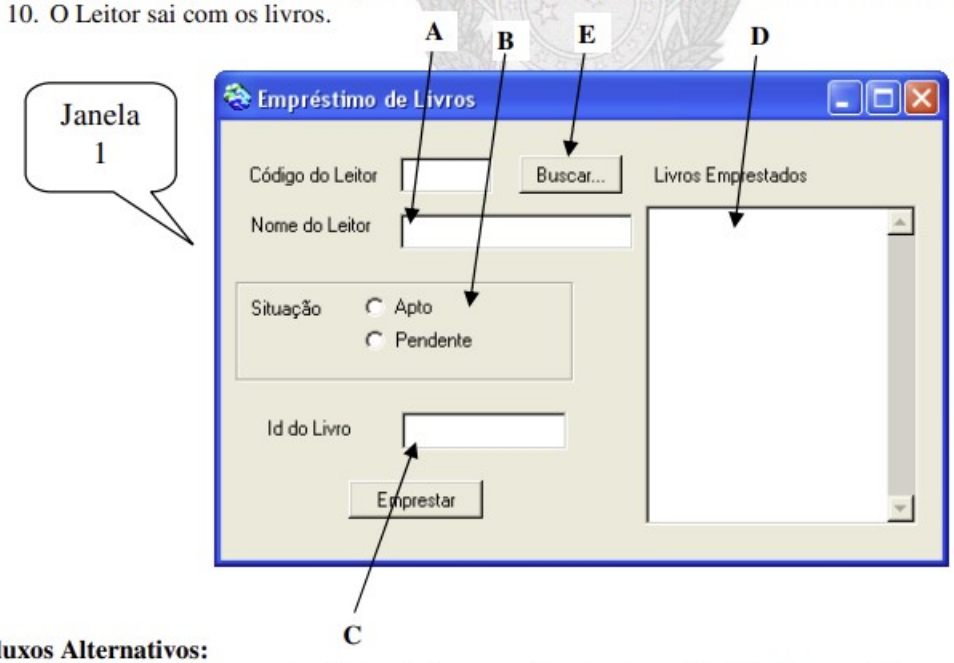
Caso de Uso: Emprestar Livro

Ator Principal: Atendente

. . .

Cenário de Sucesso Principal:

1. O Leitor chega ao balcão de atendimento da biblioteca e diz ao atendente que deseja emprestar um ou mais livros da biblioteca.
2. O Atendente seleciona a opção "Realizar um empréstimo" no menu principal do sistema de biblioteca.
3. O Atendente solicita ao leitor sua carteira de identificação, seja de estudante ou professor.
4. O Atendente passa a caneta leitora de código de barras na carteira de identificação.
5. O Sistema exibe nos campos A e B da Janela 1 o nome do leitor e sua situação.
6. O Atendente solicita os livros a serem emprestados.
7. Para cada um deles, o atendente lê o ISBN do livro por meio de seu código de barras, e o sistema exibe o ISBN no campo C da Janela 1. o Atendente clica no botão Emprestar para concretizar o empréstimo.
8. O Sistema exibe no campo D da Janela 1 o nome do livro e sua data de devolução.
9. O Atendente passa os livros pelo desmagnetizador para que possam sair da biblioteca.
10. O Leitor sai com os livros.



Fluxos Alternativos:

(1-8). A qualquer momento o Leitor informa ao Atendente que desistiu do empréstimo.

3. O Leitor informa ao Atendente que esqueceu a carteira de identificação.
   1. O Atendente usa o botão E da Janela 1 para fazer uma busca pelo cadastro do Leitor. Pede ao leitor seu CPF para garantir que ele é mesmo quem diz ser. Caso não coincida o CPF, cancela a operação.

. . .

Figura 6 - Caso de Uso Completo para "*Emprestar Livro*" (versão concreta).

Observe que o formato concreto completo possui detalhes que só são conhecidos mais à frente no processo de desenvolvimento, pois requer detalhes e decisões de projeto que são tomadas nas fases mais avançadas de elaboração e construção do PU. Já o formato abstrato completo pode ser escrito bem antes, pois se pode manter a descrição o mais geral possível, possibilitando que as decisões que envolvem detalhes desconhecidos sejam tomadas posteriormente.

1.7 – Relacionamentos entre casos de uso

Os casos de uso podem ter diversos tipos de relacionamentos uns com os outros. Por exemplo, pode ser que a execução de um caso de uso implique na execução de outro. Ou pode ser que um caso de uso possua uma parte que se repete em outros casos de uso. Para evitar redundância de texto pode-se isolar essas partes em casos de uso separados, e relacioná-los uns aos outros.

1.7.1 – O Relacionamento de Inclusão

Quando um caso de uso possui um comportamento parcial comum a vários outros casos de uso, pode-se criar um caso de uso separado para a parte comum e utilizar o relacionamento “incluir” entre eles. Esse relacionamento indica obrigatoriedade, ou seja, quando diz-se que um primeiro caso de uso inclui um segundo, necessariamente o segundo é executado quando o primeiro o é.

No diagrama de casos de uso em UML, o relacionamento de inclusão é feito por meio de uma seta tracejada entre os casos de uso, direcionada para o caso de uso incluído, e com o estereótipo «include» sobre a linha traçada, conforme mostrado na Figura 7. Neste exemplo, tanto as operações de saque quanto de depósito em uma conta bancária incluem o registro da movimentação da conta.

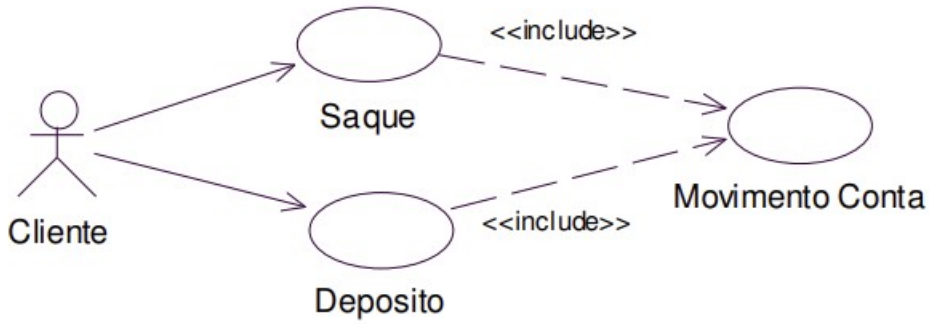


Figura 7 – Relacionamento “include”.

Na descrição do caso de uso, o relacionamento de inclusão pode ser mostrado por meio de uma referência ao caso de uso incluído, seja no cenário de sucesso principal ou nos fluxos alternativos. Por exemplo, para o caso de uso Saque da Figura 7, que inclui o caso de uso "Movimento Conta", a descrição poderia ser a da Figura 8.

Caso de Uso: Saque

...

Cenário de Sucesso Principal:

1. O Cliente chega ao balcão do Caixa e solicita uma quantia em dinheiro.
2. O Cliente entrega ao Caixa seu cartão magnético.
3. O Caixa escolhe a opção de saque e passa o cartão magnético pela leitora.
4. O Sistema exibe o nome do cliente e uma mensagem informando se seu saldo é suficiente para sacar a quantia solicitada.
5. Caso haja saldo suficiente, o Caixa entrega o dinheiro ao cliente.
6. Incluir o caso de uso: **Movimento Conta**.
7. O Cliente sai com o dinheiro.

...

Figura 8 - Caso de Uso com relacionamento *Incluir*.

1.7.2 – O Relacionamento de Extensão

Muitas vezes é indesejável modificar um caso de uso para atender situações novas que podem surgir durante a análise do sistema. Pode-se desejar acrescentar algum comportamento ao caso de uso sem, no entanto, modificar o texto original. Nesses casos o relacionamento de extensão pode ser útil. Um caso de uso estende outro se ele adiciona comportamento ao caso de uso base. Quando um fluxo alternativo é complexo e merece maior detalhamento, pode-se escrevê-lo na forma de uma extensão ao caso de uso base. Por exemplo, se o Leitor da Biblioteca esqueceu sua carteira de identificação e, portanto, precisa ser identificado de outra forma, pode-se fazer um caso de uso para tratar essa ocorrência e ligar esse caso de uso ao caso de uso "Emprestar Livro" por meio de um relacionamento de extensão, usando o estereótipo «extend», conforme ilustrado na Figura 9.

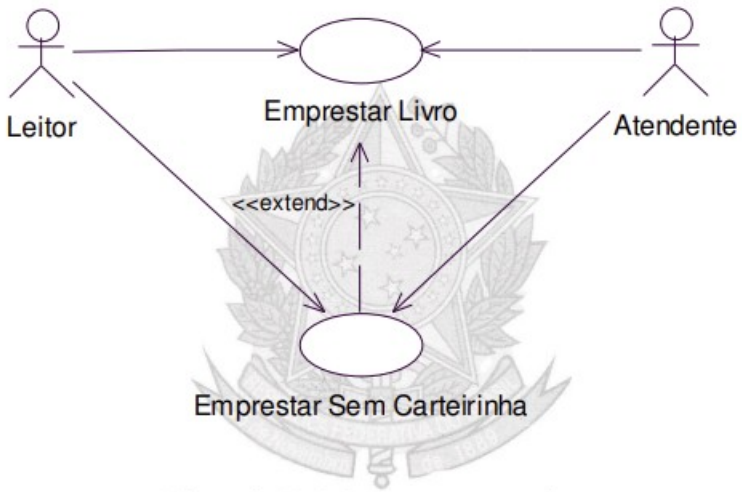


Figura 9 - Relacionamento *Extend*.

2 – Ferramentas CASE para apoiar os casos de uso

Diversas ferramentas CASE (Computer-Aided Software Engineering – Engenharia de Software Auxiliada por Computador) estão disponíveis para dar suporte à UML. Além da funcionalidade de desenhar os diagramas, imprimí-los, exportá-los para outros editores, etc., uma ferramenta CASE oferece validações e consistência entre modelos e geração automática de código a partir de certos diagramas. Por exemplo, a partir do diagrama de classes (que será apresentado mais adiante neste curso), pode-se gerar o código fonte em Java ou C++ de cada classe e de seus métodos canônicos. Métodos canônicos são os métodos básicos que, idealmente, toda classe deve possuir, como por exemplo, um método para criar objetos da classe, para destruir um objeto, para atribuir e recuperar valor de cada um de seus atributos, entre outros.

Dentre as ferramentas CASE mais utilizadas atualmente, podemos citar a JUDE Community, Rational Rose, a Visual Paradigm, a ArgoUML, a Together e a Poseidon. A ArgoUML é livre, podendo ser modificada para atender necessidades específicas. Porém, possui algumas limitações de desempenho. As demais ferramentas citadas devem ser licenciadas, mas existem versões demonstrativas que podem ser obtidas pela Internet. Neste curso será utilizada a ferramenta Rational Rose, de propriedade da IBM, que disponibiliza um programa acadêmico para convênio com Universidades e versões de avaliação pela Internet.

Para os casos de uso, a ferramenta JUDE Community oferece um editor de diagramas de caso de uso e também oferece do lado inferior esquerdo da tela, um espaço chamado “Definition” onde se pode colocar a descrição textual do caso de uso.

3 – Como fazer o planejamento para as próximas iterações?

Depois que a fase de Concepção estiver finalizada e um conjunto inicial de Casos de Uso tiver sido especificado, deve-se fazer um planejamento antes de iniciar a fase de Elaboração (já que essa fase é feita em iterações). Esse planejamento consiste, basicamente, em determinar quantas iterações serão feitas na Elaboração e quais os casos de uso que serão desenvolvidos em cada iteração. De acordo com a filosofia do PU, os casos de uso de maior risco para o sistema devem ser atacados logo nas primeiras iterações. Dessa forma, outra tarefa importante na fase de Concepção é elaborar uma tabela de planejamento como a Tabela 3.

Obviamente, muitas vezes é difícil planejar exatamente como serão as próximas iterações de forma a não fugir do que foi planejado. Entretanto, deve-se fazer esse planejamento para se ter uma idéia do que se espera ao fim de cada iteração. A idéia é uqe cada iteração cuide de uma funcionalidade do sistema. Por exemplo, para um sistema de hotel pode-se esperar que ao final da primeira iteração tenha-se a funcionalidade de reservas completa. Na segunda iteração a funcionalidade de hospedagem e na terceira a funcionalidade de tratamento de consumo dos hóspedes.

Na Tabela 3 tem-se uma simulação em que na primeira iteração cinco casos de uso serão desenvolvidos, na segunda quatro, na terceira apenas dois e na última cinco casos de uso. Outra diretriz do PU é que os casos de uso mais simples, como relatórios e consultas sejam deixados para as últimas iterações.

Tabela 3 – Modelo de Tabela para Planejamento de Iterações

	Iterações	*Caso de Uso*
Elaboração	1ª Iteração	Caso de Uso 1
		Caso de Uso 2
		Caso de Uso 3
		Caso de Uso 4
		Caso de Uso 5
	2ª Iteração	Caso de Uso 6
		Caso de Uso 7
		Caso de Uso 8
		Caso de Uso 9
	3ª Iteração	Caso de Uso 10
		Caso de Uso 11
	4ª Iteração	Caso de Uso 12
		Caso de Uso 13
		Caso de Uso 14
		Caso de Uso 15
		Caso de Uso 16

Referências

[Alexander 77] Christopher Alexander et. al., A Pattern Language, Oxford University Press, New York, 1977.
[Alexander 79] Christopher Alexander, The Timeless Way of Building, Oxford University Press, New York, 1979.
[Appleton 97] Appleton, Brad. Patterns and Software: Essential Concepts and Terminology, disponível na WWW na URL: http://www.cmcrossroads.com/bradapp/docs/patterns-intro.html
[Bauer, 2004] Bauer, Christian; King, Gavin. Hibernate in Action, Manning Publications.
[Beck 87] Beck, Kent; Cunningham, Ward. Using Pattern Languages for Object-Oriented Programs, Technical Report nº CR-87-43, 1987, disponível na WWW na URL: http://c2.com/doc/oopsla87.html
[Booch, 1995] Object Solutions : Managing the Object-Oriented Project (Addison-Wesley Object Technology Series by Grady Booch , Pearson Education; 1st edition (October 12, 1995)
[Buschmann 96] Buschmann, F. et al. A System of Patterns, Wiley, 1996. [Coad 92] Coad, Peter. Object-Oriented Patterns. Communications of the ACM, V. 35, nº9, p. 152-159, setembro 1992.
[Coad 95] Coad, P.; North, D.; Mayfield, M. Object Models: Strategies, Patterns and Applications, Yourdon Press, 1995. [Coleman et al, 1994] Coleman, Derek et al. Object Oriented Development - the Fusion Method, Prentice Hall, 1994.
[Coplien 92] Coplien, J.O. Advanced C++ Programming Styles and Idioms. Reading-MA, Addison-Wesley, 1992.
[Coplien 95] Coplien, J.; Schmidt, D. (eds.) Pattern Languages of Program Design, Reading-MA, Addison-Wesley, 1995. [Deitel, 2002] Deitel, Harvey M; Deitel Paul J. JAVA – como programar, Editora Bookman, 4a Edição.
[Gamma 95] Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. Design Patterns - Elements of Reusable Object-Oriented Software. Reading-MA, Addison-Wesley, 1995.
[Goodwill, 2002] Goodwill,J. Martering Jakarta Struts, Wiley Publishing, Inc. [Krutchen, 2000] Krutchen, Philippe. The Rational Unified Process, An Introduction, Second Edition, Addison Wesley, 2000.
[Larman, 2004] Larman, Craig. Utilizando UML e Padrões, 2a edição, Bookman, 2004. [Rational, 2000] RATIONAL, C. Unified Modeling Language. Disponível na URL: http://www.rational.com/uml/references, 2000.
[Rumbaugh, 1990] Object-Oriented Modeling and Design by James R Rumbaugh, Michael R. Blaha, William Lorensen, Frederick Eddy, William Premerlani, Prentice Hall; 1st edition (October 1, 1990)
[Waslawick, 2004] Waslawick, Raul. Análise e Projeto de sistemas de Informação Orientados a Objetos, Campus, 2004.