



ANÁLISE E PROJETO ORIENTADO A OBJETOS

Professores:

Ana Paula Gonçalves Serra

André Luiz Ribeiro

Keity Yamamoto

Modelo de Casos de Uso

O modelo de casos de uso é um dos modelos da UML, composto do diagrama de casos de uso que apresenta um conjunto de casos de uso, atores e relacionamentos, e de um detalhamento do diagrama de casos de uso, geralmente uma descrição textual.

O modelo de casos de uso é construído durante as discussões entre os desenvolvedores do sistema de software e os envolvidos (clientes, usuários, áreas da empresa envolvidas diretamente ou indiretamente no sistema) para especificar os requisitos.

Um modelo de casos de uso apresenta os requisitos funcionais pretendidos do sistema e os elementos externos que interagem com o sistema.

1. Objetivo Principal do Modelo de Casos de Uso

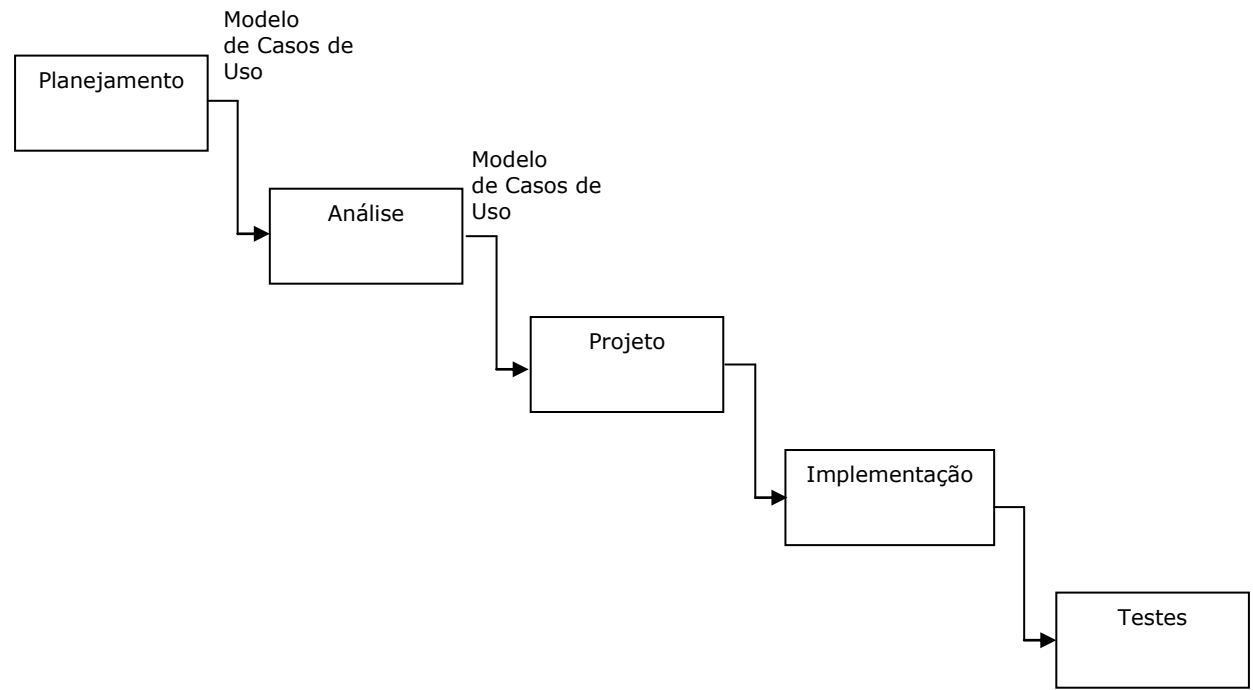
Especificar, visualizar, construir e documentar o comportamento que cada parte do sistema deve ter.

Os casos de uso descrevem os OBJETIVOS do sistema e não as FUNÇÕES!

Detalhando o objetivo principal, obtêm-se objetivos específicos que são:

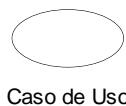
- Descrever os requisitos funcionais do sistema, de acordo, com todos os envolvidos e os desenvolvedores do sistema de software;
- Fornecer uma clara e consistente descrição do escopo do sistema e o que ele deve fazer;
- Fornecer requisitos para elaboração do diagrama de classes;
- Simplificar alterações e extensões (manutenção) do sistema e verificar o impacto através dos casos de uso afetados e novos;
- Fornecer cenários para validação dos testes do sistema;
- Definir o perfil dos usuários que irão utilizar o sistema;
- Definir as condições de início, término e exceções de cada função do sistema;
- Validar outros diagramas da UML.

O diagrama de Casos de Uso é elaborado na fase de Planejamento e/ou Análise através da ferramenta CASE. O diagrama auxilia também nas fases de Projeto, Implementação e Testes.

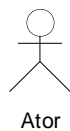


2. Notação do Diagrama de Caso de Uso

O diagrama de casos de uso é composto basicamente por:



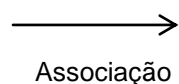
Caso de uso: Um caso de uso é um conjunto de ações que o sistema realiza para produzir um resultado observável para um ator. (Objetivos do sistema)



Ator: são elementos externos ao sistema, que interagem com os casos de uso. Consideram-se atores “alguém” ou “alguma coisa” que não faz parte do sistema, mas que interage com ele.

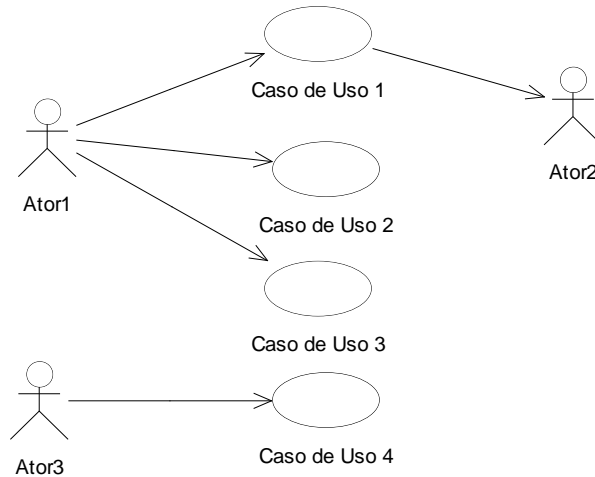
É o papel executado por alguém e não a pessoa em si. Por exemplo: José é o Administrador do sistema e também usuário comum. Os atores serão o Administrador e Usuário e não a pessoa José.

Exemplos de atores: usuários diretos do sistema, sistemas externos que irão interagir com o sistema ou hardware específico.

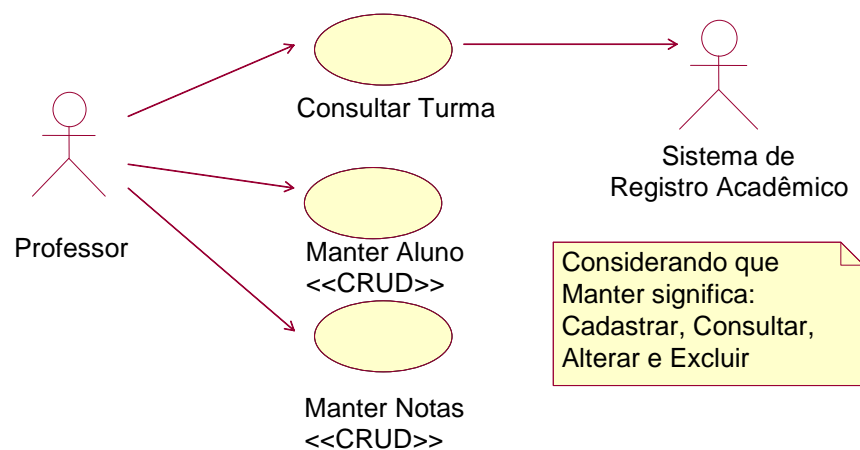


Associação é um relacionamento entre o ator e o caso de uso.

Exemplo da Notação de Diagrama de Caso de Uso:



Exemplo de Diagrama de Caso de Uso:



3.1. Notação Complementar do Diagrama de Caso de Uso

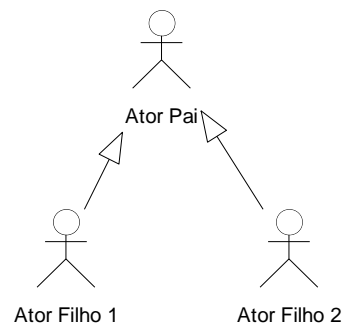
- **Relacionamento entre atores**

Atores podem ter um relacionamento de generalização, representando que possuem características e ações comuns, além de algumas características e ações adicionais diferenciadas.

A generalização é representada por uma linha contínua e uma seta em formato de triângulo. A seta em formato de triângulo deve ficar apontada para o ator pai (com características e ações comuns) e a outra parte da linha no ator filho que herda as características e ações comuns do pai e ainda possui características e ações específicas.

Uma generalização de ator de um tipo de ator (descendente) para outro tipo de ator (ascendente) indica que o descendente herda o papel que o ascendente pode desempenhar em um caso de uso.

Exemplo da Notação de Generalização:



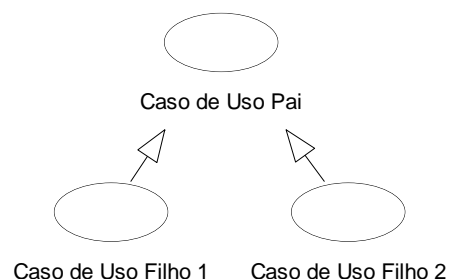
- **Relacionamento entre casos de uso**

Os casos de uso podem ter relacionamentos entre si. Essas relações devem ser analisadas somente APÓS A ESPECIFICAÇÃO DOS CASOS DE USO, pois com a descrição detalhada dos casos de uso podemos verificar possíveis relacionamentos entre eles.

Basicamente existem três tipos de relacionamento entre casos de uso, que são:

- ❖ **Generalização:** Como no relacionamento de generalização entre atores, o relacionamento de generalização entre casos de uso é utilizado quando existem dois ou mais casos de uso que têm comportamento, estrutura e finalidade comuns. Quando isso ocorre, pode descrever as partes compartilhadas em um caso de uso novo, que é especializado pelos casos de uso filho.

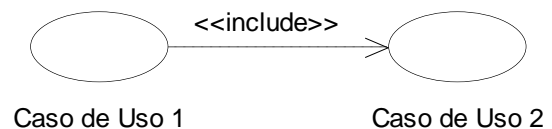
Exemplo da Notação de Generalização:



- ❖ **Inclusão (*Include*):** Relacionamento que indica que um caso de uso contém comportamento definido em um outro caso de uso, e toda vez que o caso de uso base for executado, o caso de uso de inclusão relacionado pelo <<include>> também será executado. O relacionamento de inclusão pode ser usado para:

- Separar o comportamento do caso de uso base que não seja necessário para compreender a finalidade principal do caso de uso, apenas o resultado é importante.
- Separar o **comportamento que seja comum a dois ou mais casos de uso**.

Exemplo da Notação de Inclusão (include):

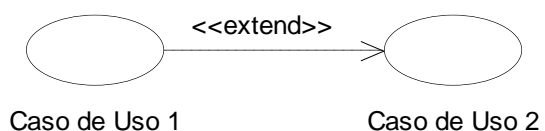


- ❖ **Extensão (Extend):** Relacionamento que especifica que o comportamento de um caso de uso base pode ser estendido para um outro caso de uso adicional. A extensão é condicional, o que significa que sua execução depende do que tiver acontecido durante a execução do caso de uso base. O relacionamento de extensão pode ser utilizado para:
 - Mostrar que uma parte de um caso de uso é um comportamento opcional (ou possivelmente opcional) do sistema. Isso faz a diferenciação entre comportamento opcional e comportamento obrigatório em um modelo.
 - Mostrar que um subfluxo só é executado em determinadas condições excepcionais.
 - Mostrar que pode haver um conjunto de segmentos de comportamento dentre os quais um ou vários podem ser inseridos em um ponto de extensão de um caso de uso base. Os segmentos de comportamento que são inseridos (e a ordem na qual são inseridos) dependerão da interação com os atores durante a execução do caso de uso base.
 - Podem representar os fluxos alternativos relevantes de um determinado caso de uso.

Princípio da Extensão:

- Ele **sempre** adiciona comportamento a um caso de uso.
- O caso de uso base deve continuar intacto e tendo valor por si próprio.
- O comportamento básico do caso de uso estendido deve sempre ficar intacto

Exemplo da Notação de Extensão (extend):



3. Passo a Passo para Elaboração do Diagrama de Caso de Uso

A seguir são apresentados os passos que podem ser realizados para elaboração do diagrama de caso de uso.

Passo 1. Identificação dos Atores

Listar os possíveis usuários diretos do sistema, os sistemas externos e hardwares específicos. Para isso, pode-se identificar os possíveis atores e objetivos em relação ao sistema (atores x objetivo).

Dicas para encontrar atores:

- Inicie identificando os atores.
- Trabalhe do específico para o geral.
- Não esqueça os atores de suporte.
- Considere as informações que você possui (documento de Visão, por exemplo).
- Lembre que atores nem sempre são pessoas.
- Foque na fronteira do sistema.
- Identifique as fontes de informação.
- Evolua os atores junto com os casos de uso.

Perguntas para ajudar:

- Quem vai fornecer, usar ou remover informações?
- Quem usará essa funcionalidade?
- Quem está interessado em um determinado requisito?
- Em que parte da organização o sistema é usado?
- Quem vai dar suporte e manter o sistema?
- Quais são os recursos externos do sistema?
- Que outros sistemas precisarão interagir com este?

Passo 2. Identificação dos Casos de Uso

Listar os possíveis OBJETIVOS do sistema. Para identificar esses objetivos, algumas questões podem ser feitas:

- Quais objetivos o ator necessita do sistema? O que o ator precisa fazer?
- O que o ator precisa fazer? O ator precisa calcular, consultar, criar, destruir, modificar, ou registrar algum tipo de informação? Por quê?
- O ator deve ser notificado sobre eventos do sistema ou precisa notificar o sistema sobre algo? O que estes eventos representam em termos de funcionalidade?
- Existem objetivos de comunicação com outros sistemas?
- O ator precisa estar informado sobre certas ocorrências no sistema?
- Que informações devem ser modificadas ou criadas no sistema?

Complementando :

- Não se preocupe com partes comuns (pelo menos no início).
- Não confundir casos de uso com “Funções”.

- Foque SEMPRE no valor para o ator.
- Não esqueça dos casos de uso de suporte e operacionais.
- Evolua os casos de uso junto com os atores.

Passo 3. Associação entre os atores e casos de uso

Elaboração propriamente dita do diagrama de caso de uso. Deve-se verificar se todo ator possui pelo menos uma associação com um caso de uso e se todo caso de uso interage com algum ator.

4. Descrição do Diagrama de Casos de Uso

Além da notação gráfica o diagrama de casos de uso deve possuir uma especificação de detalhamento do diagrama de caso de uso. O modelo padrão do RUP (*Rational Unified Process*) é que será utilizado nas aulas, e aborda os seguintes itens.

- Catálogo de atores, contendo o nome do ator e uma descrição de cada ator representado no diagrama.
- Nome do caso de uso.
- Breve descrição do caso de uso.
- Fluxo Básico (detalhar passo a passo do caso de uso).
- Fluxos Alternativos (qualquer exceção que ocorra nos passos do fluxo básico, deve ser detalhado como fluxo alternativo).
- Pré-Condições (lista de condições que têm que ser verificadas antes que o caso de uso comece).
- Pós-Condições (lista de condições que têm que ser verificadas depois do fim do caso de uso).

Dicas para escrever casos de uso:

- Escreva na voz ativa. “O Sistema valida a quantidade entrada” em vez de “A quantidade entrada deve ser validada pelo sistema”.
- Escreva na voz presente.
- Descreva as interações que ocorrem entre Atores e o Sistema.
- Use subfluxos para simplificar descrições complexas.
- Não encha seu modelo de casos de uso com o CRUD(Create, Read, Update, Delete).
- Não tenha medo de capturar os detalhes.

5. Considerações e Observações:

- Um caso de uso sempre deve ter um relacionamento com um ator. Isso implica que cada caso de uso deve ter associações de comunicações com os atores. O motivo dessa regra é forçar o sistema a fornecer apenas a funcionalidade de que os usuários precisam e nada

mais. Ter casos de uso que ninguém solicita é uma indicação de que algo está errado no modelo de casos de uso ou nos requisitos. Entretanto, há algumas exceções a essa regra:

- Um caso de uso filho em um relacionamento de generalização não precisa ter um ator associado a ele se o caso de uso pai está associado a um ator.
- Um caso de uso pode ser iniciado de acordo com uma programação (por exemplo, uma vez por semana ou uma vez por dia), o que significa que o relógio do sistema é o iniciador. O relógio do sistema é interno ao sistema. E o caso de uso não é iniciado por um ator, mas por um evento do sistema interno. Se não ocorrer outra interação do ator no caso de uso, ele não terá nenhuma associação com os atores. Entretanto, para esclarecer, você pode usar um ator fictício "Tempo" para mostrar como o caso de uso é iniciado nos diagramas de casos de uso.
- Na grande maioria das modelagens de diagramas casos de uso a notação básica (ator, relacionamento e caso de uso) é suficiente para especificar os requisitos. Isso significa que não necessariamente é obrigatório utilizar herança, <<include>> e <<extends>> em toda modelagem de sistemas, inclusive a notação de <<include>> e <<extends>> pode ser omitida no diagrama e detalhada no passo a passo do fluxo básico e/ou alternativo.
- Geralmente utiliza-se o termo "Manter" para indicar ações de incluir, consultar, alterar e excluir. Para isso, pode-se utilizar o estereótipo¹ <<CRUD>> (create, read, update e delete). Com isso deixamos o detalhamento das funções para a especificação do Caso de Uso e mantemos o diagrama mais claro.
- Após a elaboração do diagrama de caso de uso e das especificações dos casos de uso deve-se verificar se há a necessidade de criação de generalização entre atores e casos de uso, inclusões (include) e extensões (extend).

Copyright © 2012 - 2017 Profa. Ana Paula Gonçalves Serra, Prof. André Luiz Ribeiro e Prof. Keity Yamamoto. Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, da Profa. Ana Paula Gonçalves Serra, Prof. André Luiz Ribeiro e Prof. Keity Yamamoto.

¹ **Estereótipo:** é um mecanismo de extensão da UML que permite a metaclassificação de elementos de modelagem, com o objetivo de identificação e classificação, sendo que algumas semânticas são definidas na UML, mas novas semânticas podem ser definidas pelos envolvidos no projeto. Na UML, os estereótipos são representados por sinais duplos de maior e menor (<<estereótipo>>).