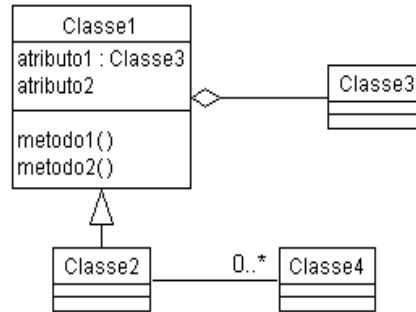


Diagramas de casos de uso

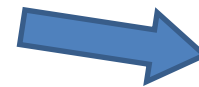
Fases de um projeto de software



**Elicitação/
Especificação
de requisitos**



Projeto



Codificação



Testes

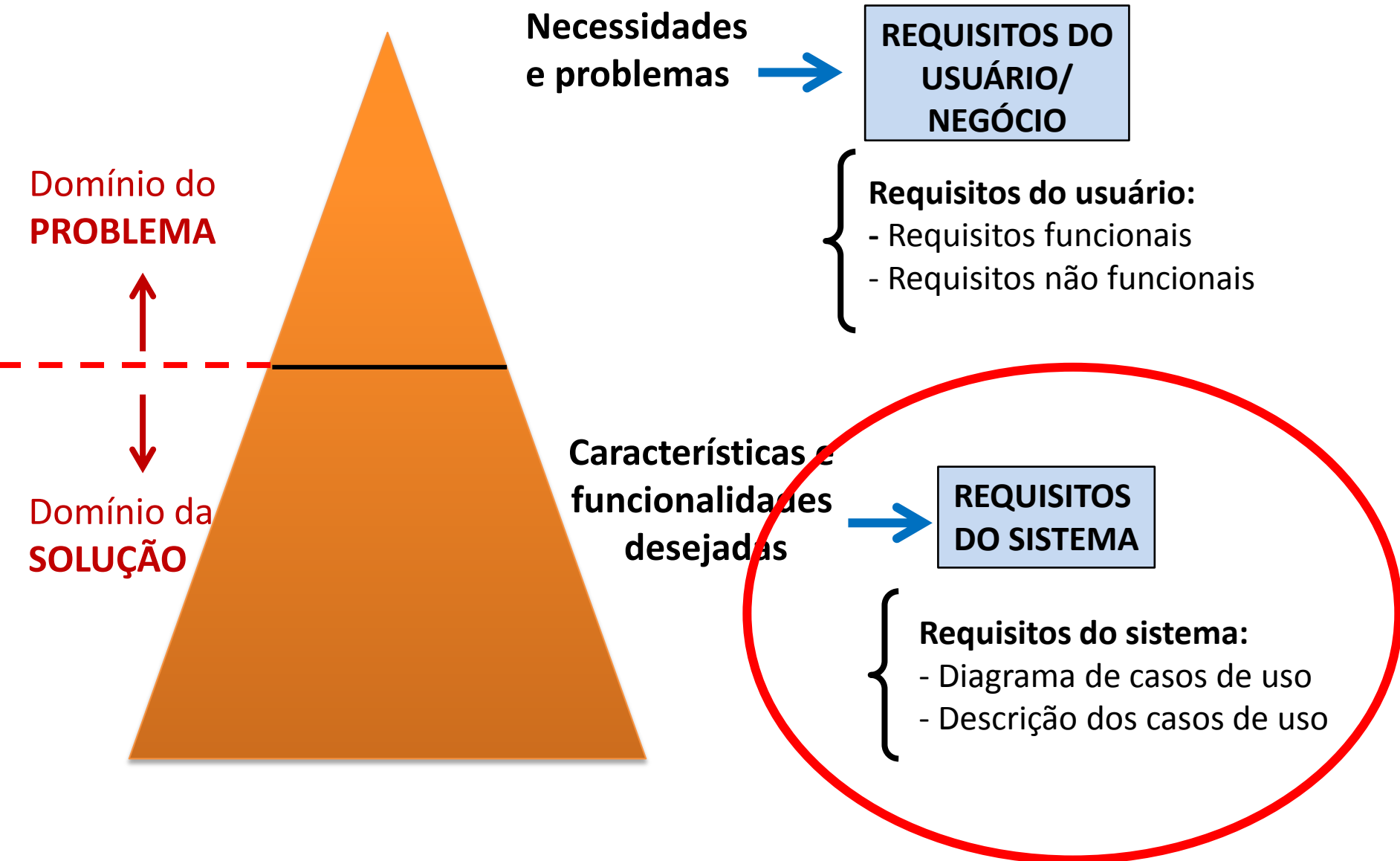


Implantação

Manutenção



Tipos de requisitos



Requisitos de sistema

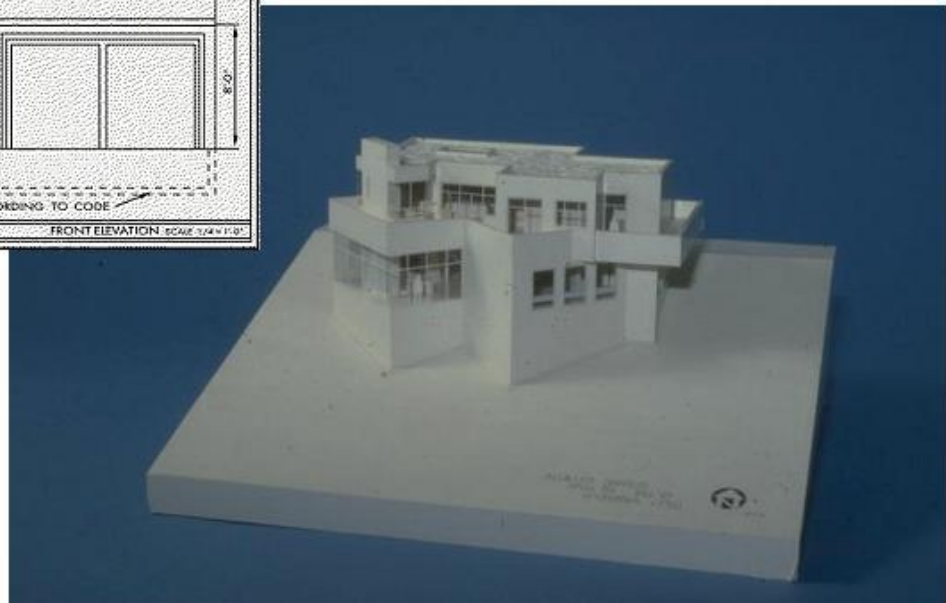
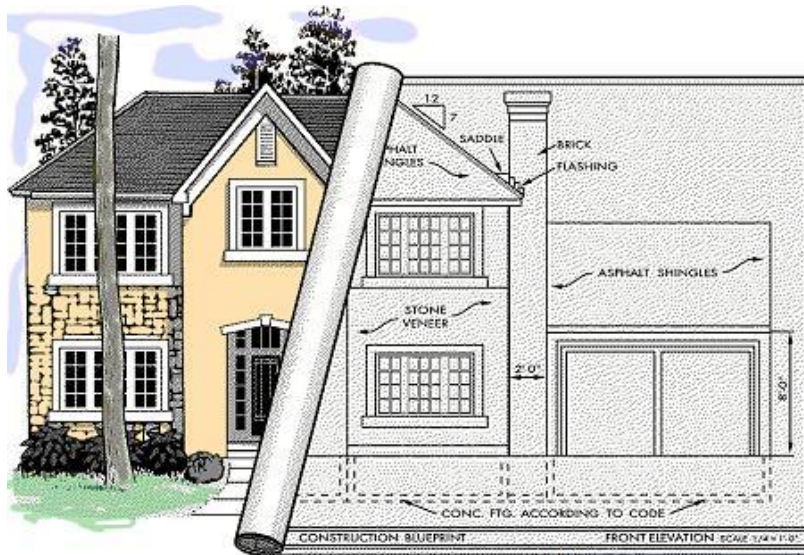
- Definem o que o sistema deve fazer (funções que deve ter)
- Procura definir o que o sistema deve ter internamente para atender adequadamente os requisitos do usuário.

UML



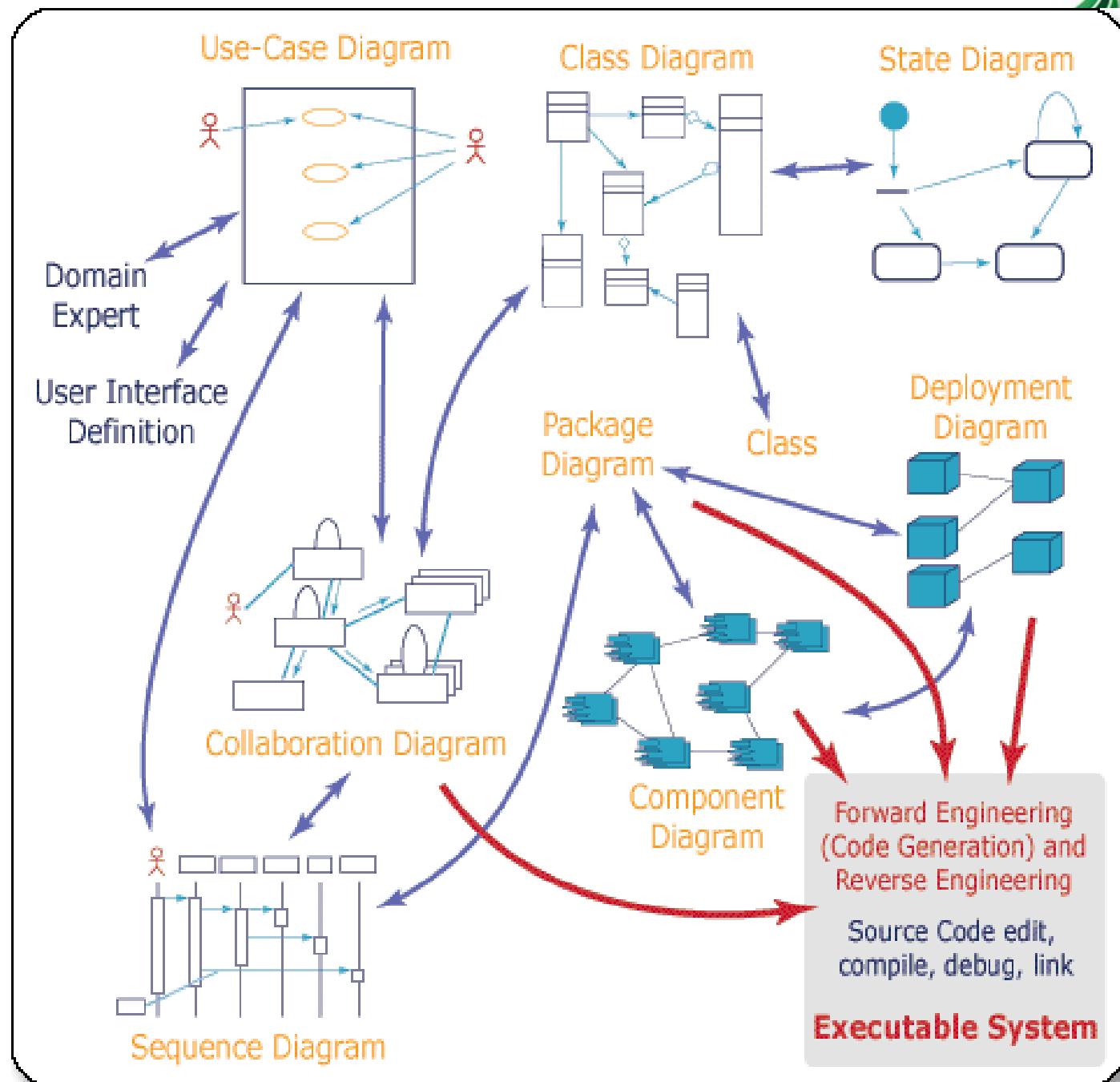
- UML – Linguagem de Modelagem Unificada
- A UML é uma linguagem visual utilizada para **modelar softwares** baseados no **paradigma de orientação a objetos**.
- Tornou-se, nos últimos anos, a linguagem **padrão de modelagem adotada internacionalmente** pela indústria de engenharia de software.

Representação do projeto de software de forma visual e padronizada, como ocorre na construção civil



Diagramas da UML

Modelagem visual:
necessidade de
modelar o sistema
segundo uma
notação padrão



UML

Unified Modelling Language



- A UML foi um passo importante em direção à padronização de desenvolvimento de software
- Surgiu em 1995 da união de três métodos de modelagem OO: Booch, OMT e OOSE
- Tornou-se padrão para modelagem e projetos de sistemas



Grady Booch
Método Booch



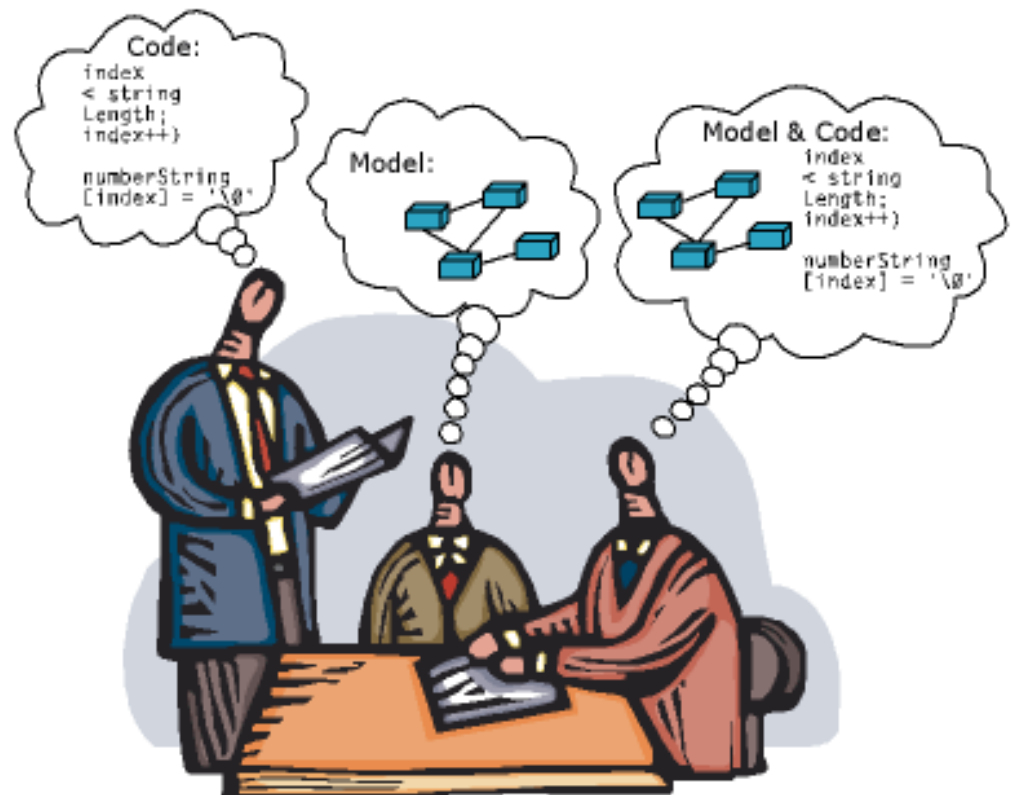
Ivar Jacobson
Método OMT (Rational)



James Rumbaugh
Método OOSE

Linguagem consistente para:

- Visualização
- Especificação
- Documentação
- Construção



Diagramas de casos de uso

- Técnica **especificação de requisitos** baseada em cenários
- identificam os agentes em uma interação (atores) e descrevem a interação com o sistema (casos de uso).

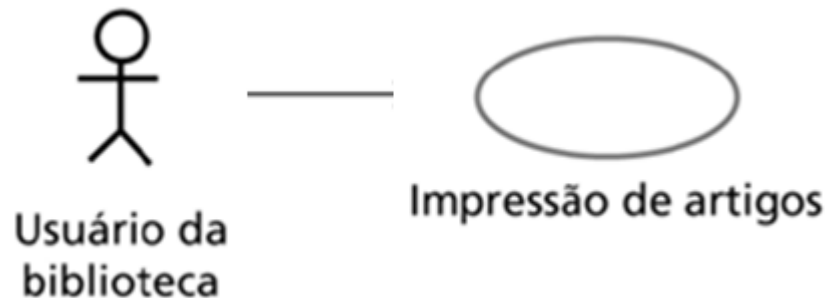
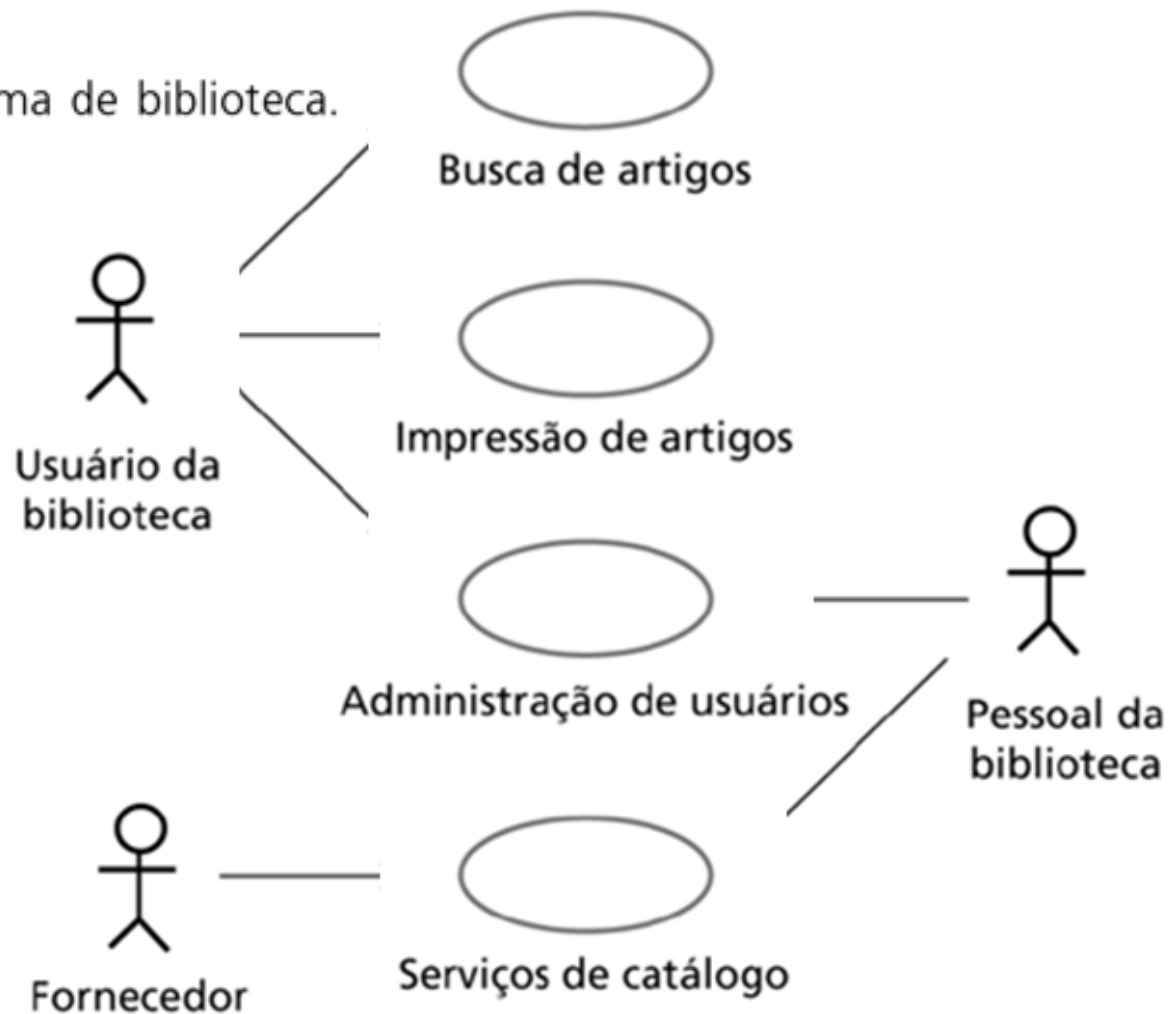


Figura 7.6

Casos de uso para o sistema de biblioteca.



Mostra os potenciais usuários do sistema (atores) e as funcionalidades que serão úteis para eles (casos de uso)

Casos de uso

- Um conjunto de casos de uso **deve descrever todas as possíveis interações do ator com o sistema.**
- Os diagramas de caso de uso **são usados para especificar os requisitos funcionais do sistema.**
- Permite que os requisitos **fiquem acessíveis e compreensíveis a todos**, por apresentarem uma **visão externa** sobre como os elementos que interagem com sistema.
- Modelagem dinâmica de sistema em **alto nível de abstração**

Principais elementos gráficos



➔ Atores

Usuário da
biblioteca



➔ Casos de uso

Busca de artigos



Usuário da
biblioteca

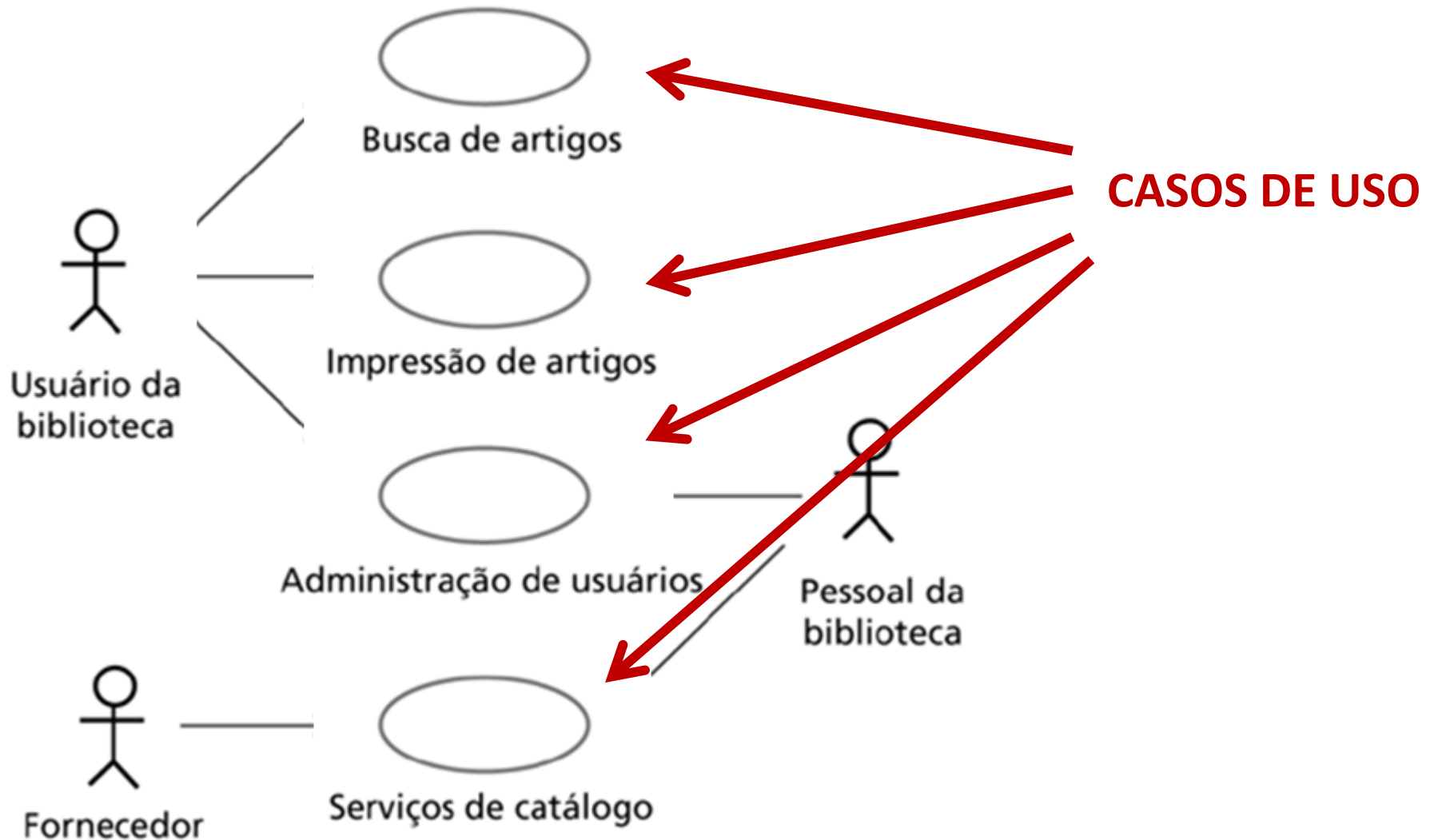


Impressão de artigos

➔ Relacionamentos entre
os elementos atores e
casos de uso

Figura 7.6

Casos de uso para o sistema de biblioteca.



Caso de uso

- Uma funcionalidade do software
- Atômica, completa (não uma fração)
- Externamente perceptível
- EX: cada uma das opções do menu de um caixa eletrônico de banco
 - *emissão de extrato de conta corrente*
 - *Realizar saque*
 - *Transferir dinheiro*

Mono sessão

- Um bom caso de uso deve ser mono sessão. Isso significa que ele deve iniciar e terminar sem ser interrompido.
- Por exemplo, o registro de uma encomenda de livros é feito em uma única sessão de uso do sistema.

Dica

Pode-se pensar assim:

- Somente será um caso de uso um **processo completo**
- Um usuário iria ao computador, ligaria o sistema, executaria o processo e em seguida poderia desligar o computador porque o processo estaria completo.

Ator

- É alguém ou algo que interage com o sistema, ou seja, é quem usa o sistema.
- Um ator pode ser um humano, outro sistema ou um software
 - *Vendedor (pessoa)*
 - *Impressora (hardware)*
 - *Sistema bancário (outro software)*



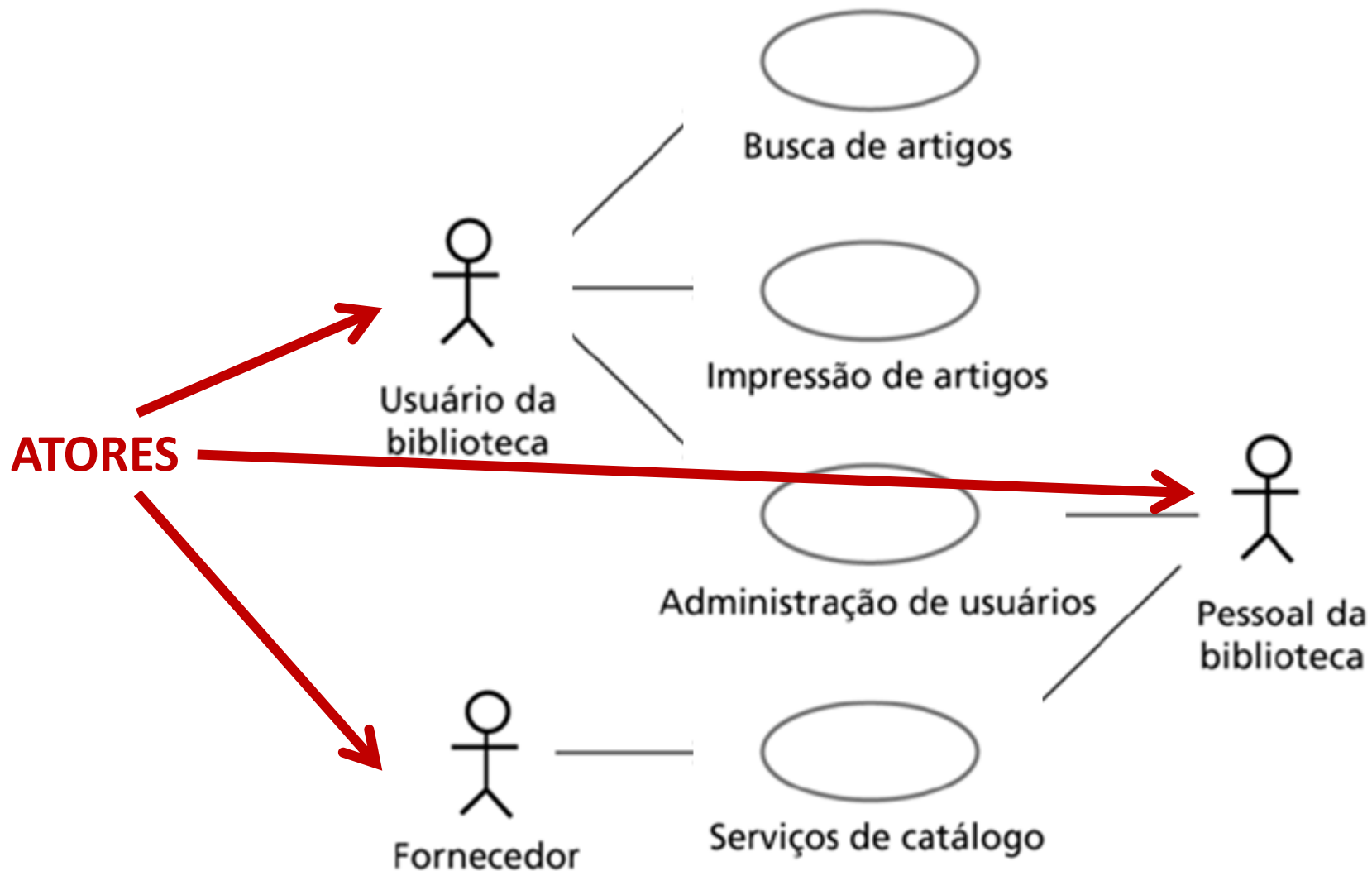
Representação de ator



Vendedor

Figura 7.6

Casos de uso para o sistema de biblioteca.



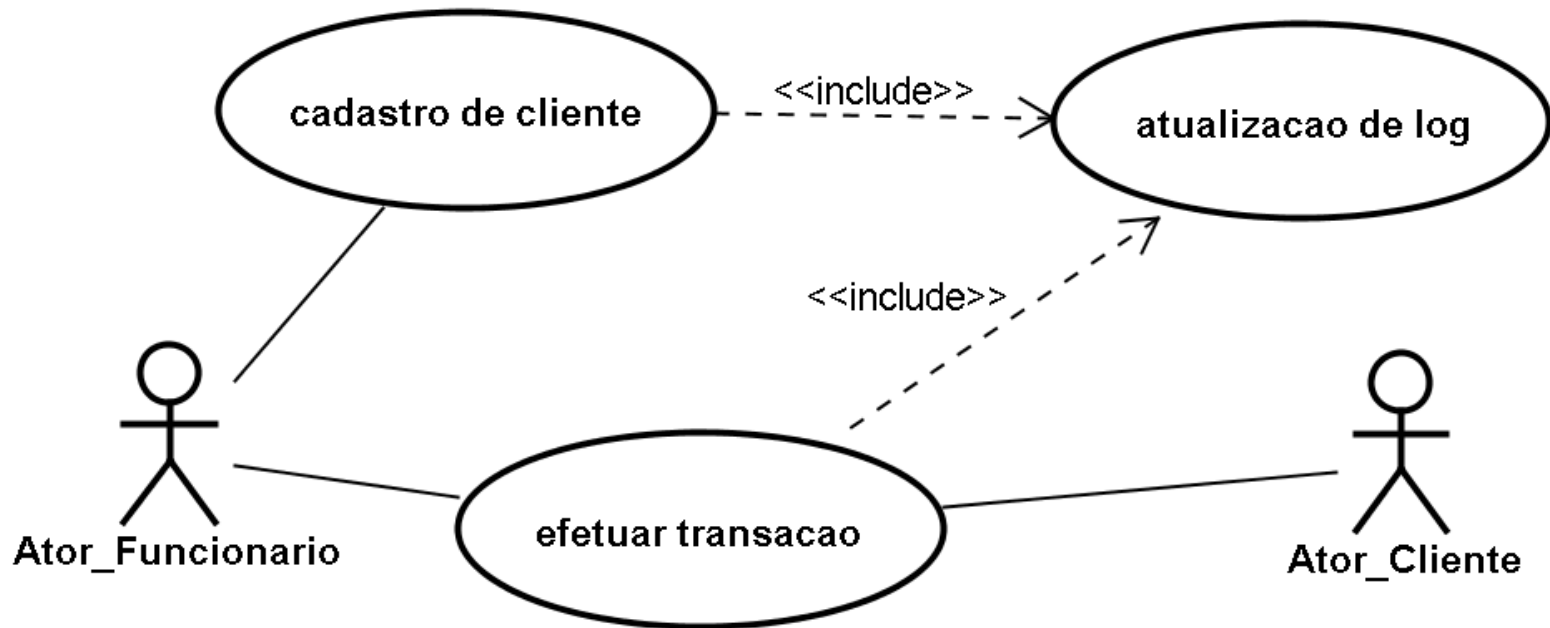
Associação entre ator e caso de uso

- Participação do ator na utilização de um caso de uso
 - Um ator sempre está associado a pelo menos um caso de uso
 - Um caso de uso sempre estará associado a pelo menos um ator

Associação entre casos de uso

Associação entre casos de uso

→ Inclusão <<include>> (obrigatório)



Explicação:

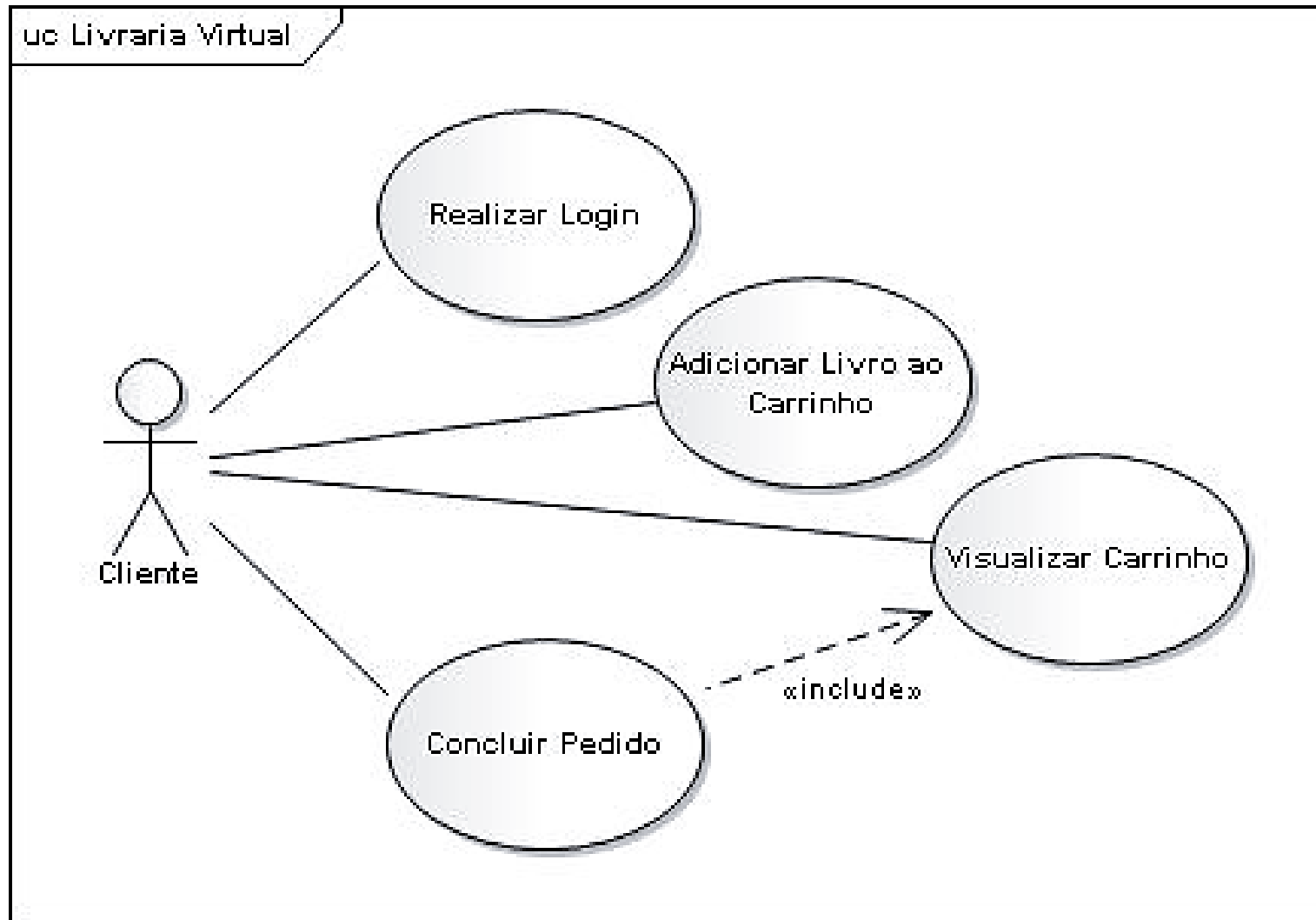
- O comportamento referente ao caso de uso *atualização de log* **está contido** tanto no caso de uso *cadastro de cliente*, quanto no *efetuar transação*

Associação de Inclusão <<include>>

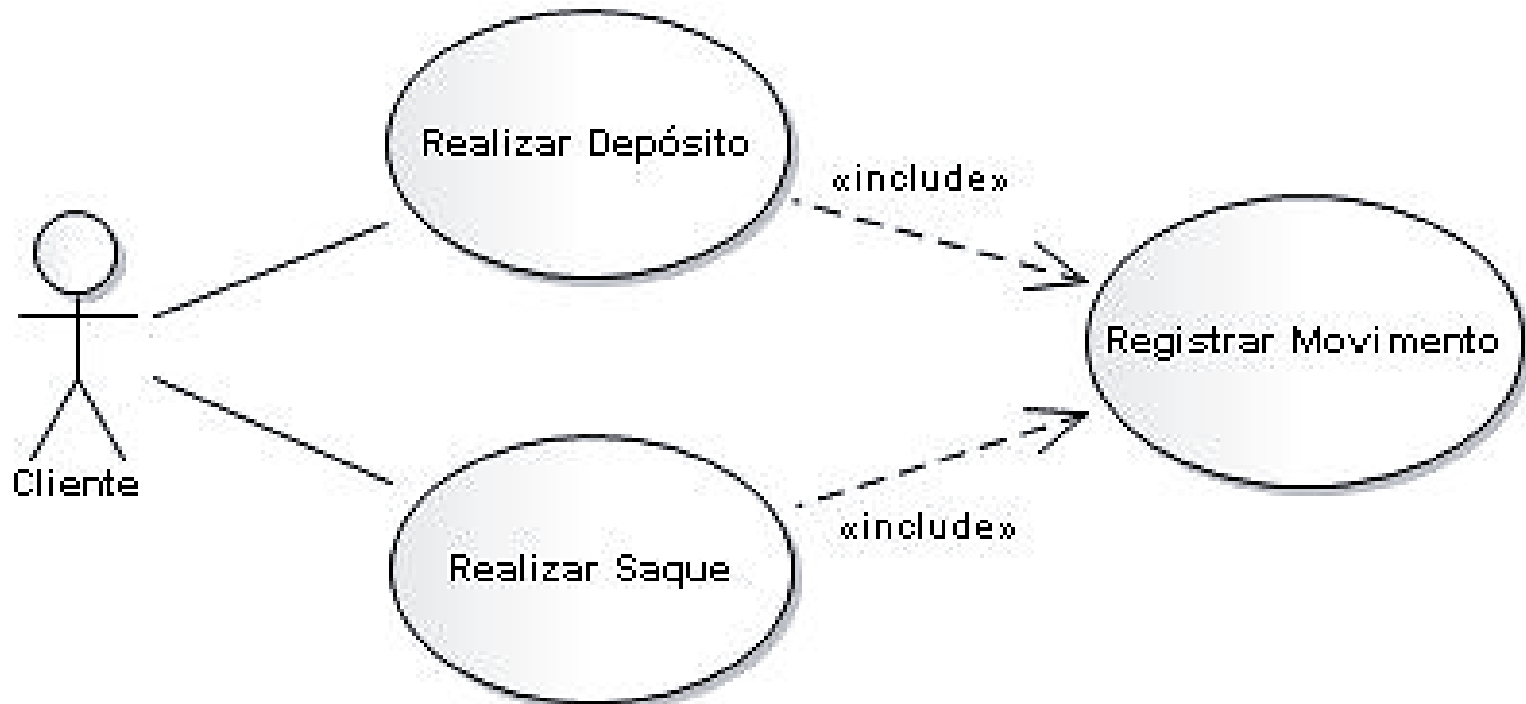


- Estabelece que parte do comportamento de um caso de uso está definida em outro caso de uso
- Um caso de uso contém o comportamento definido em outro caso de uso

Associação de Inclusão <<include>>

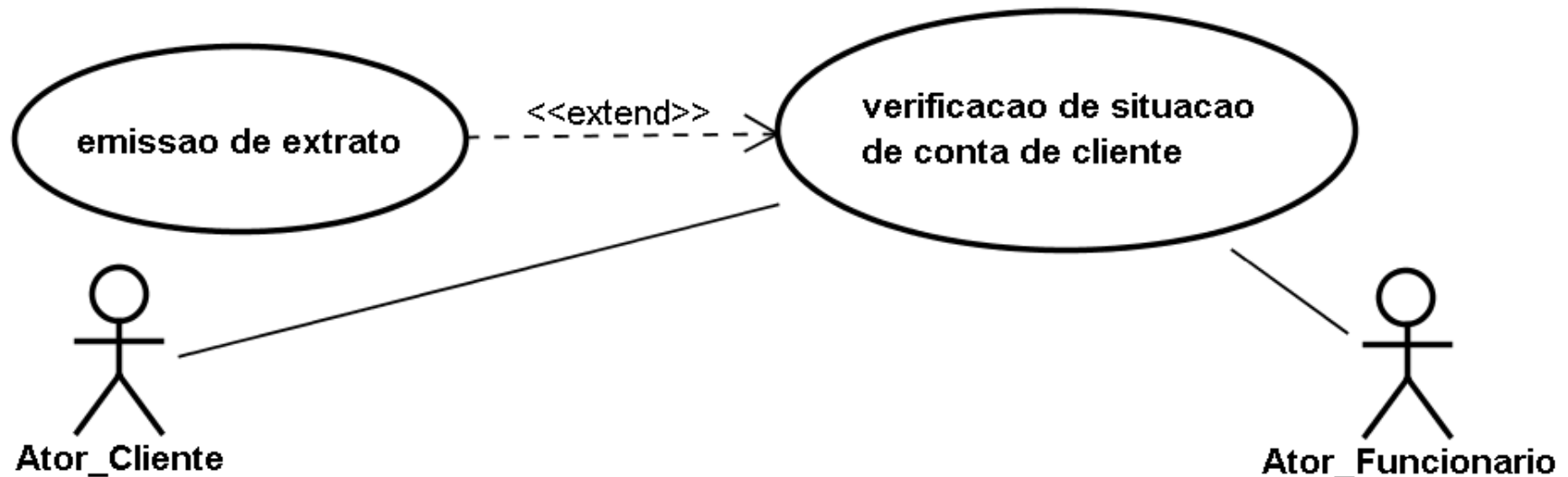


Associação de Inclusão <<include>>



Associação entre casos de uso

→ Extensão <<extend>> (opcional)



Explicação:

- *verificação de situação de conta de cliente tem seu comportamento estendido pelo caso de uso *emissão de extrato**
- Pode **ocorrer com ou sem** *emissão de extrato*

Associação de Extensão <<extend>>

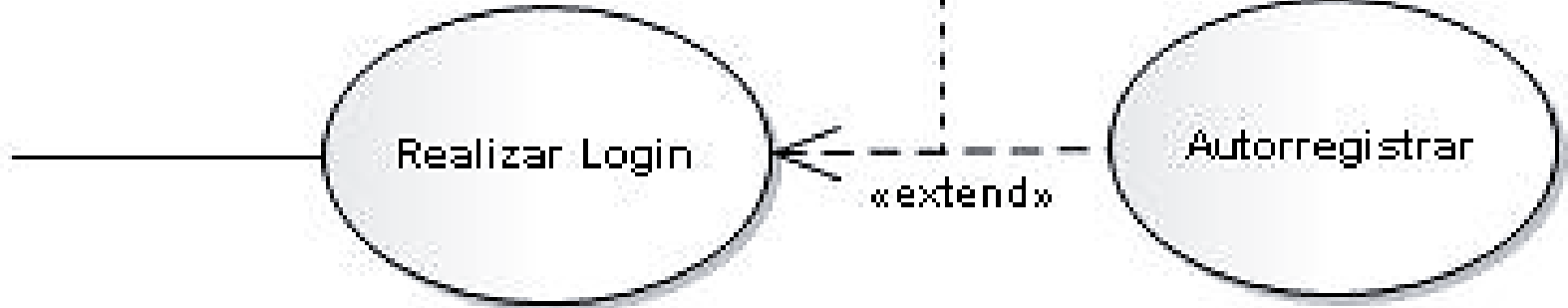
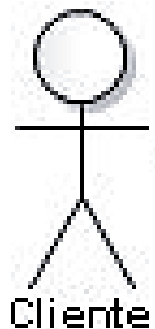


- Estabelece uma relação em que um dos casos de uso tem seu **comportamento estendido** através do comportamento definido em outro caso de uso
- Pode ou não ocorrer

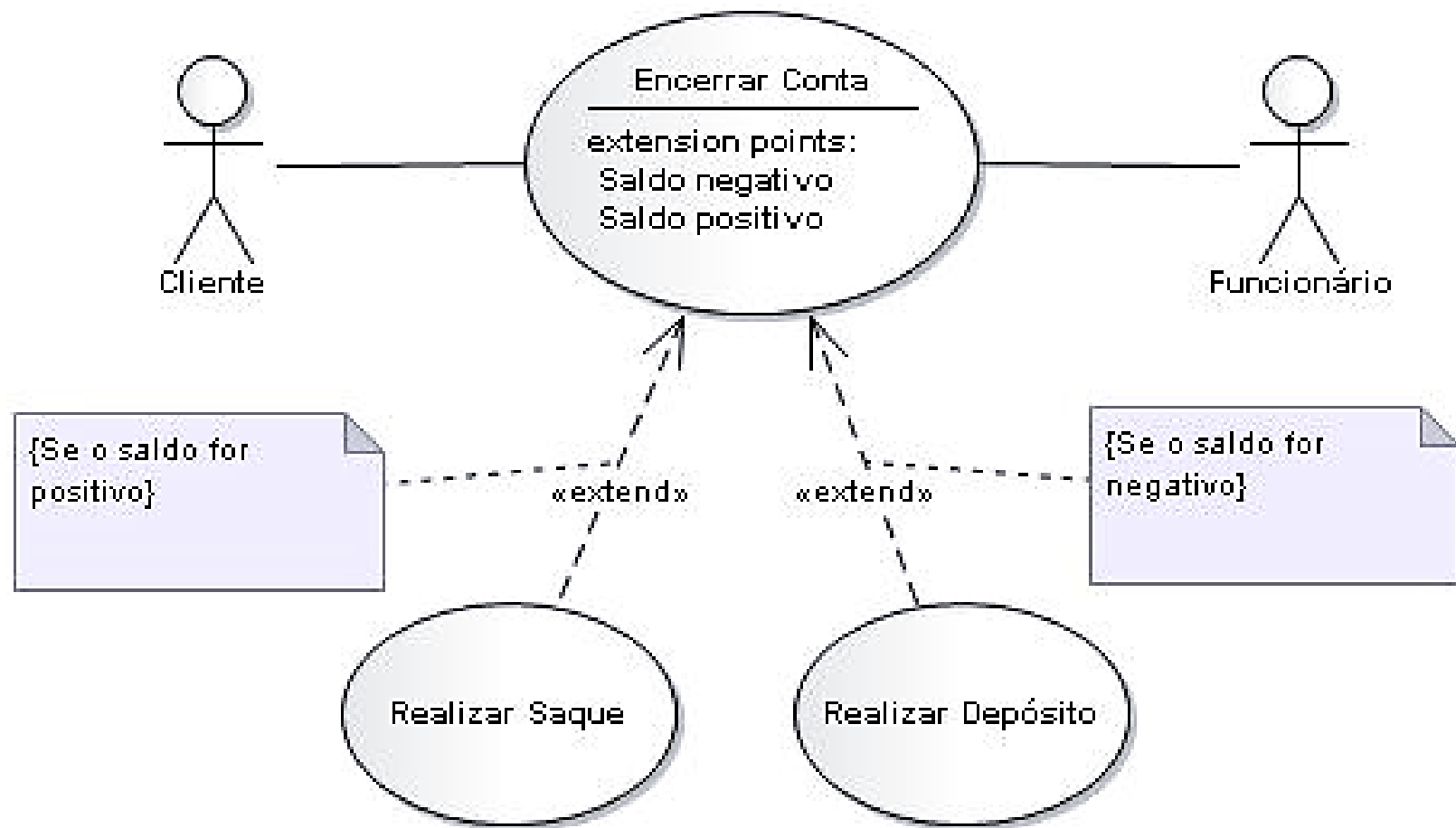
Exemplo: Extend

Restrição: define em que situação/condição deve ser executado um determinado caso de comportamento estendido

{Caso o cliente ainda não esteja registrado}



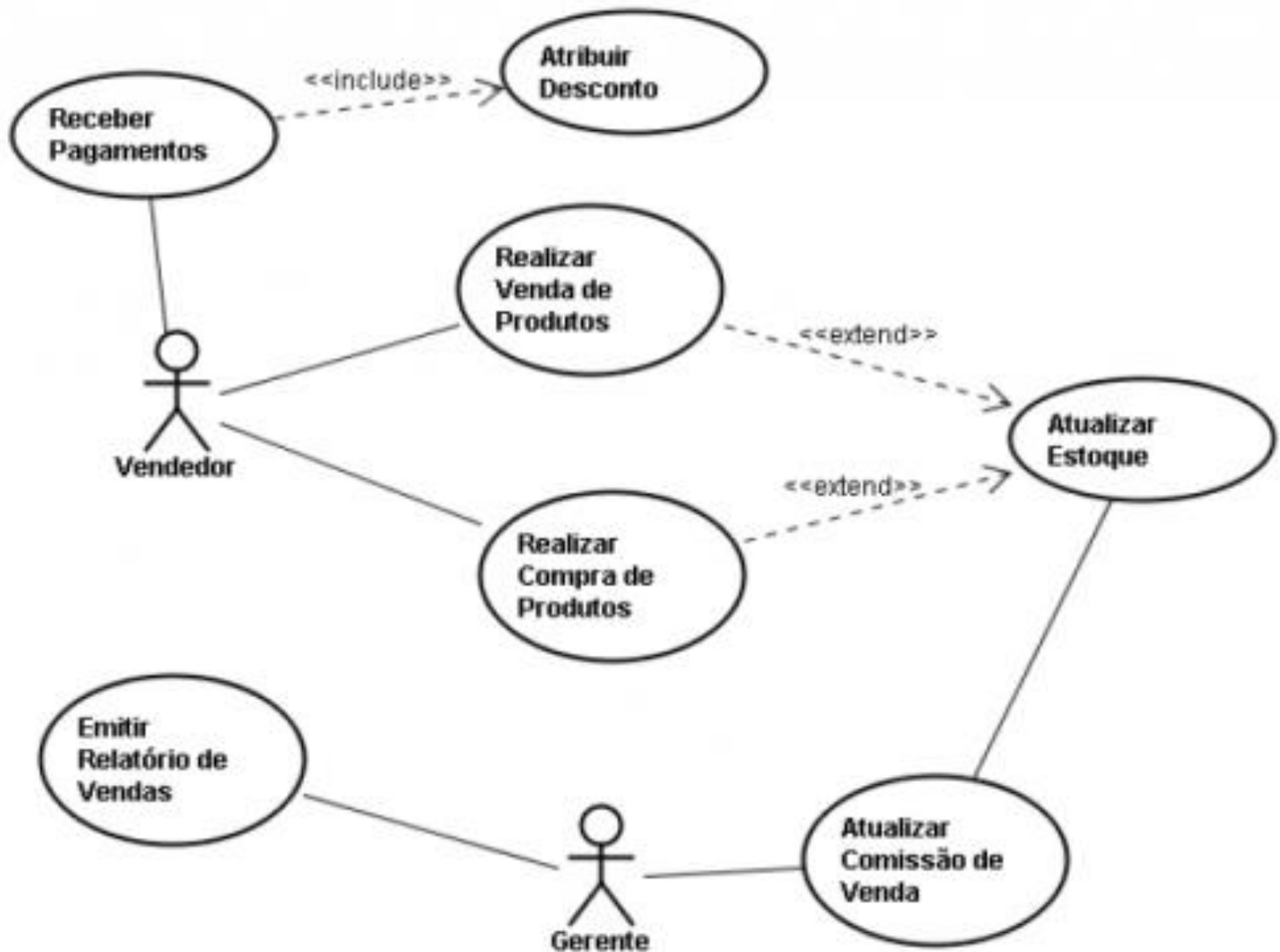
Exemplo: Extend



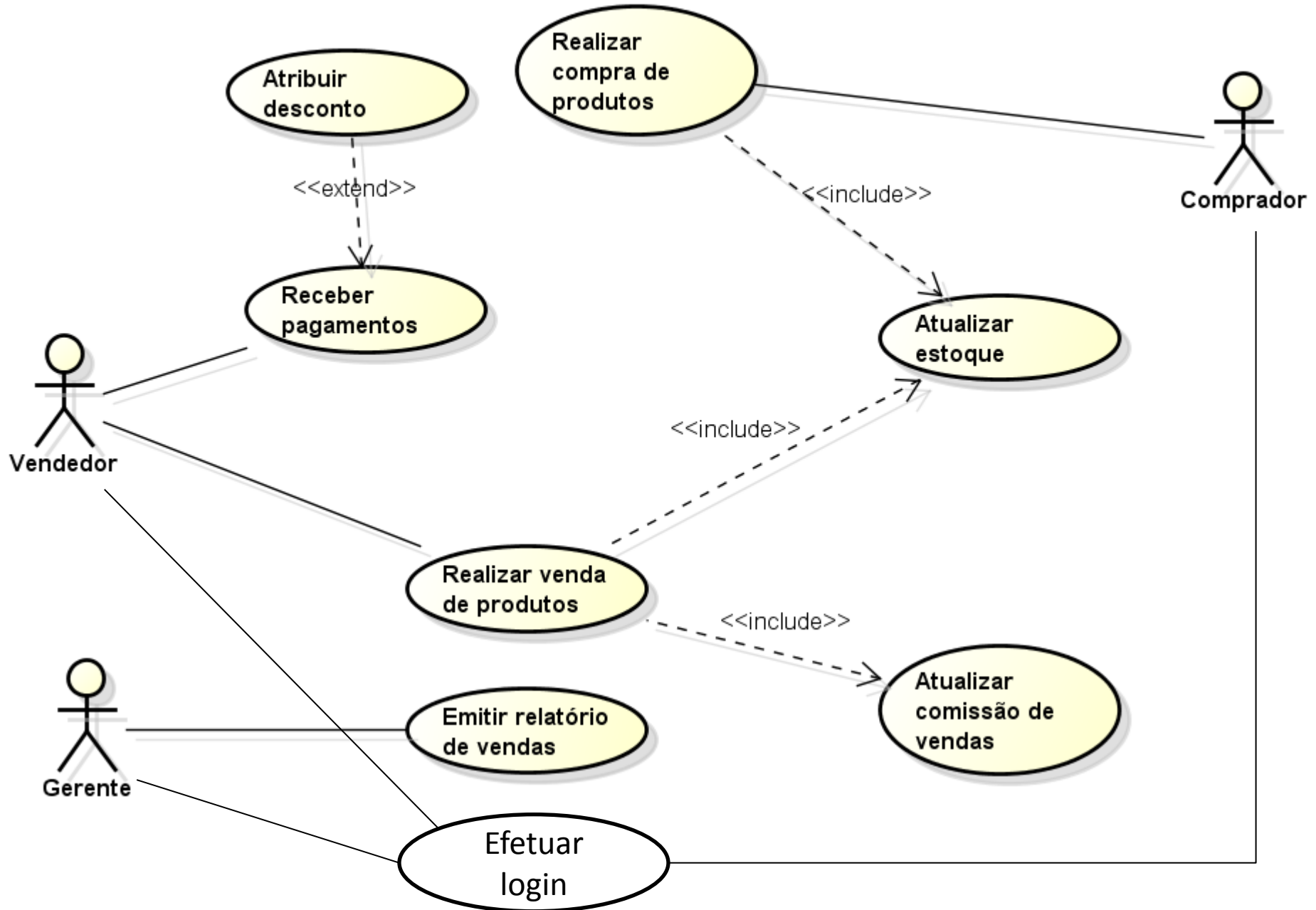
Diferença entre inclusão e extensão

- **Inclusão** denota **obrigatoriedade**
 - Quando o caso de uso principal ocorre, o caso de uso auxiliar também ocorrerá
- **Extensão** denota **opcionalidade**
 - Quando o caso de uso principal ocorre, o caso de uso auxiliar poderá ou não ocorrer

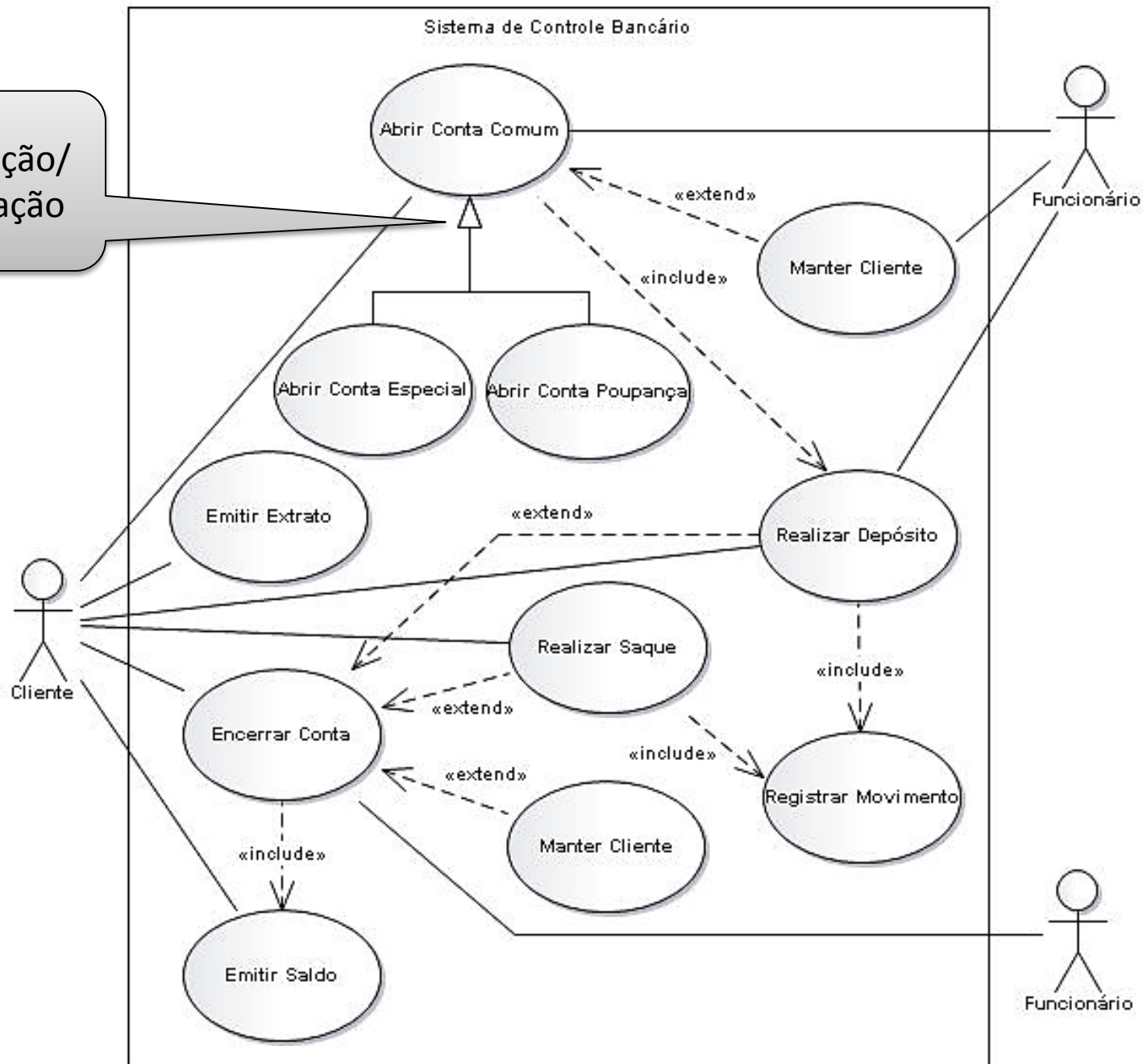
Exercício - Encontre os erros



Exercício - SOLUÇÃO



Generalização/
especialização



Software para modelagem UML

- **Astah** versão community (free):

<http://astah.net/download/>

- **Creately** (online)

<https://creately.com/>