



Conhecimento é a nossa natureza
Ciência da Computação
Fundamentos de Engenharia de Software

UML – Diagrama de Casos de Uso

Prof. Jeangrei Veiga

UML - Diagrama de Casos de Uso

- ❖ O Diagrama de Casos de Uso (Use Case) tem o objetivo de descrever um **modelo funcional** do sistema. [Deboni, 2003].
- ❖ Deve estabelecer um modelo de **requisitos funcionais** do sistema, por meio da identificação de como o mesmo é utilizado pelos elementos externos (**usuários**) e quais serviços deve oferecer (**funções**).
 - o que o sistema atual faz ou o novo sistema deve fazer
 - quem faz

UML - Diagrama de Casos de Uso

- ❖ É um dos diagramas mais importantes da construção de software orientado a objetos. Pode ser usando tanto para **levantamento de requisitos (na análise)** como para a definição das funções **de um novo sistema (no projeto)**.
- ❖ O Diagrama de Casos de Uso auxilia na comunicação entre os analistas e o cliente.
- ❖ O cliente deve ver no diagrama de casos de uso as principais funcionalidades de seu sistema.
- ❖ Muito útil para documentação dos requisitos de um sistema

UML - Diagrama de Casos de Uso

- ❖ Serve de orientação para as próximas fases
- ❖ É construído através de um processo iterativo entre os desenvolvedores do sistema e os usuários
- ❖ O modo como o sistema implementa os casos de uso (detalhes do funcionamento e dados utilizados), não é relevante na fase de identificação dos casos de uso

Diagrama de casos de uso - Exemplo

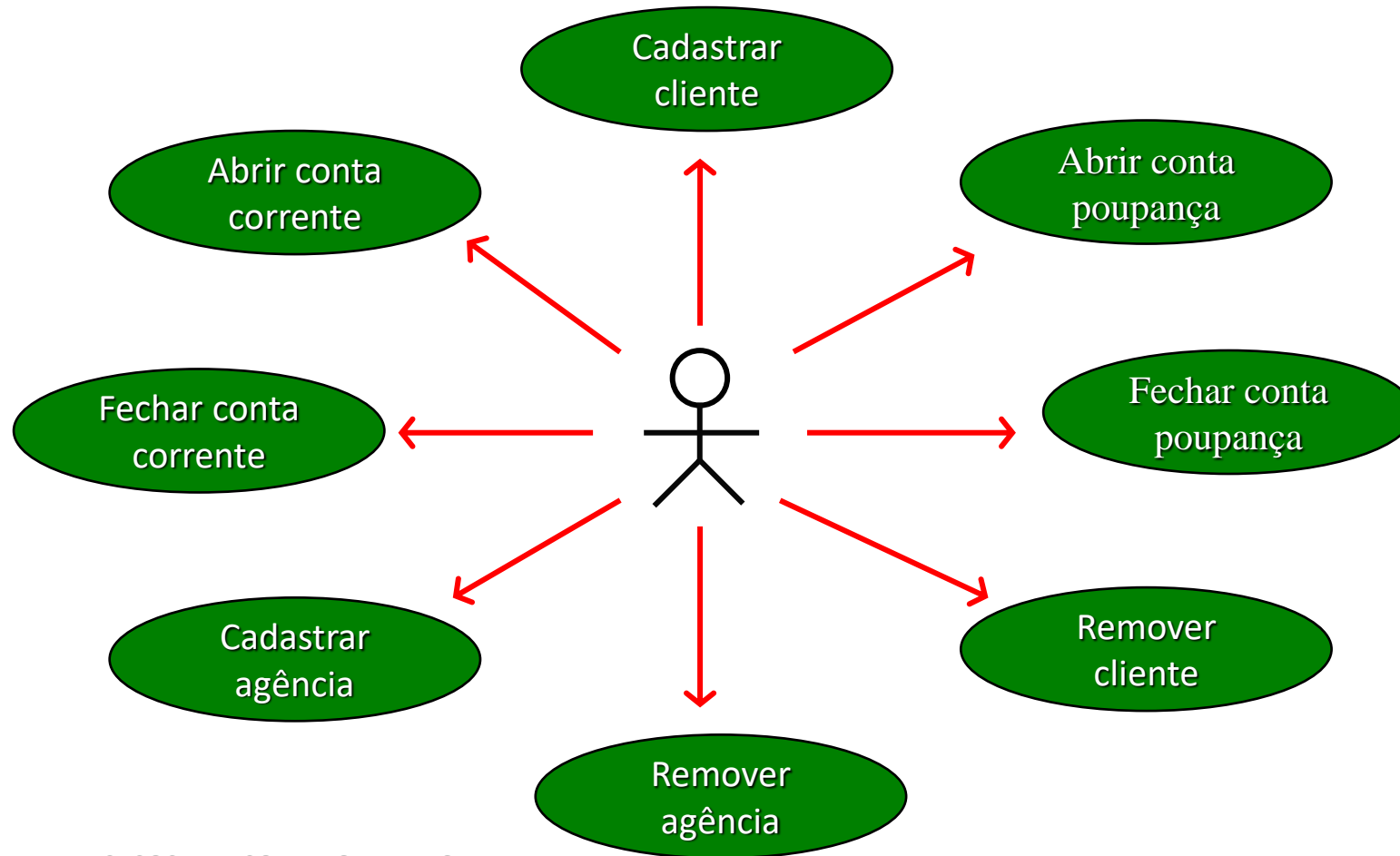


Diagrama de casos de uso (composição)

- É composto de quatro elementos básicos
 - Sistema
 - Ator
 - Caso de Uso
 - Relacionamentos / Interação

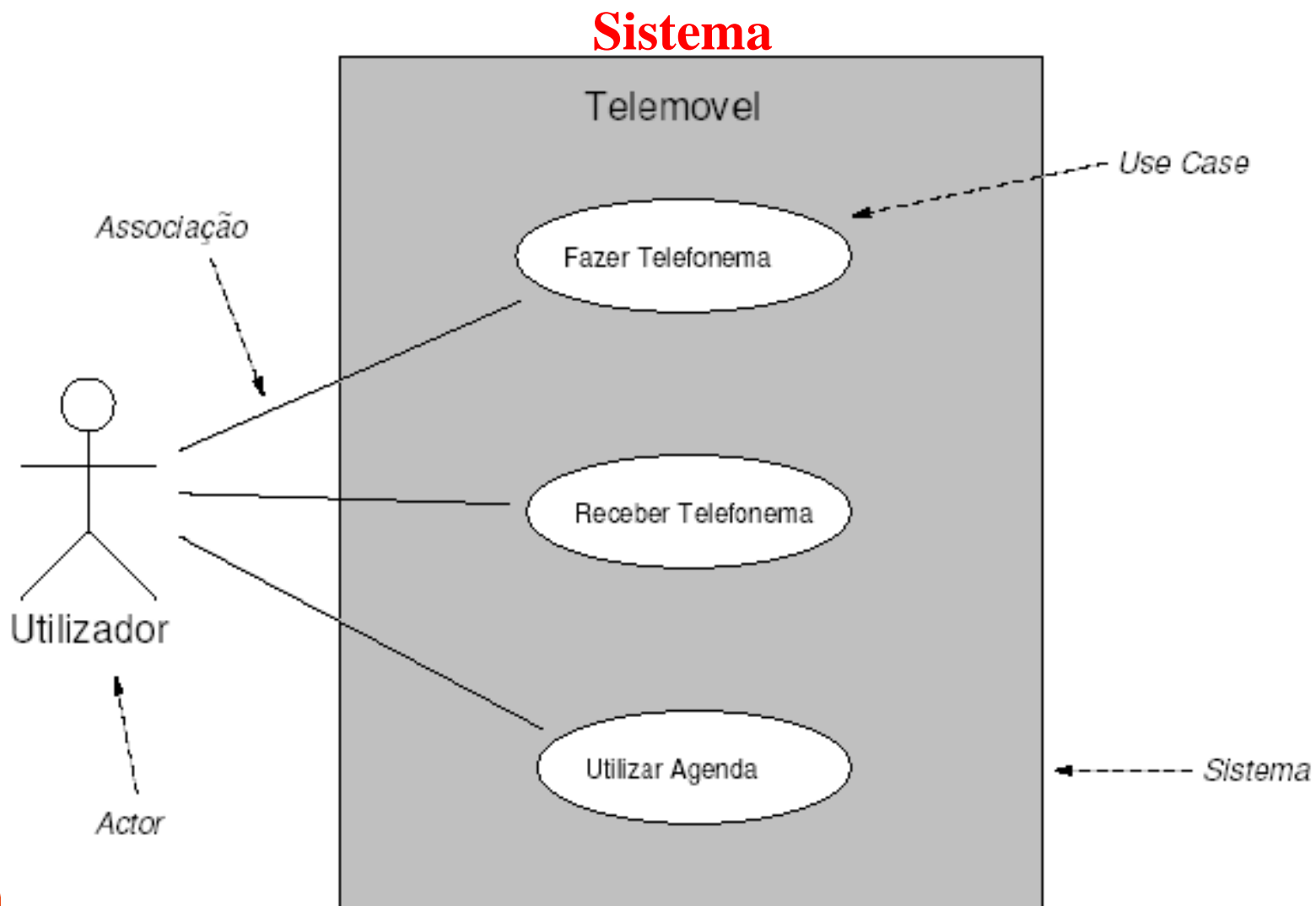
Sistema

- ❖ Significa um sistema de software, um negócio ou uma máquina
- ❖ Como parte da modelagem, os limites do sistema desenvolvido são definidos. Na modelagem, os limites do sistema podem ser representados por uma “caixa”, onde os casos de uso serão inseridos.

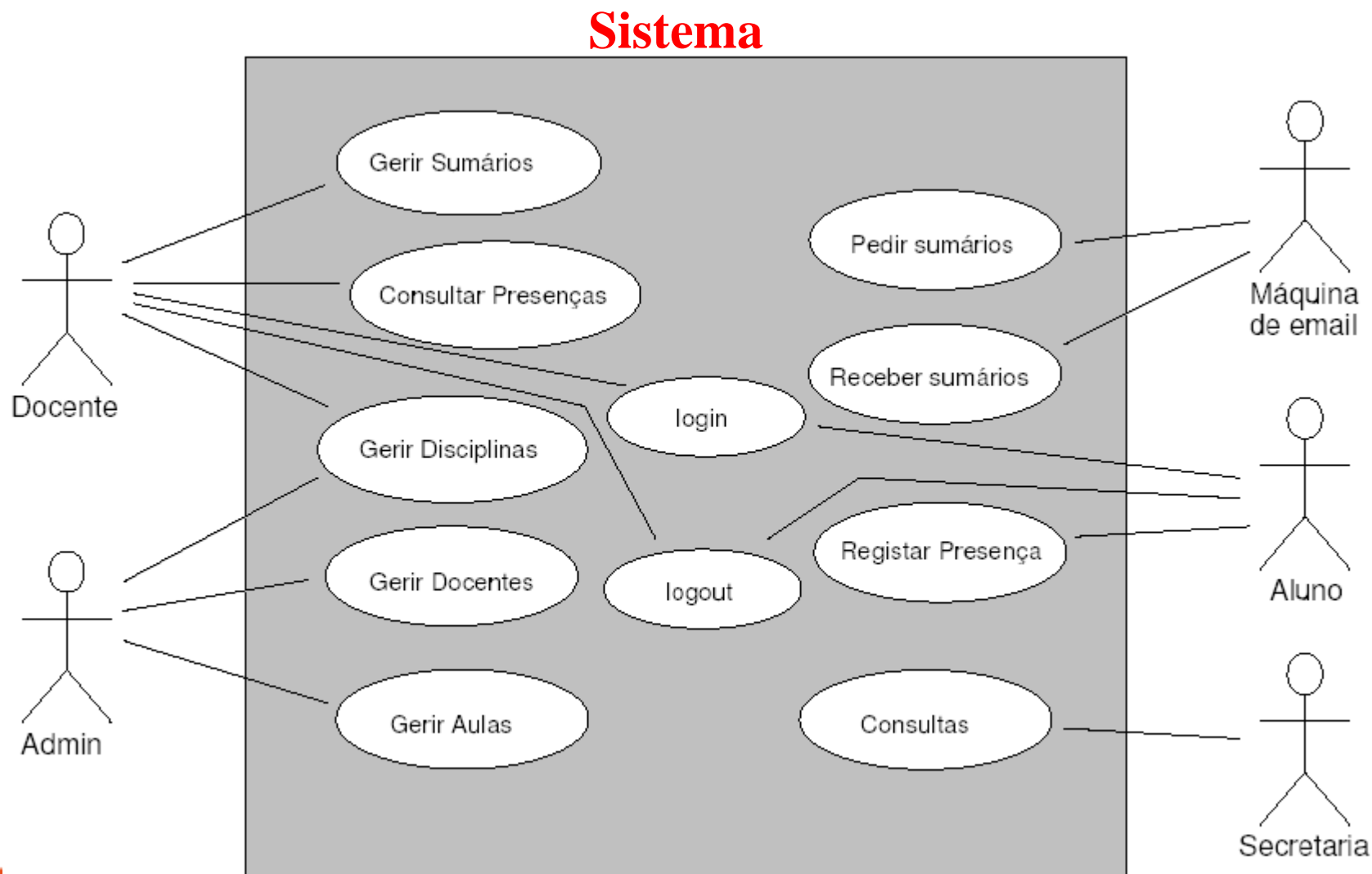
Sistema

- ❖ A fase de definição dos limites e responsabilidades de um sistema nem sempre é um processo fácil, pois nem sempre é óbvio quais tarefas são melhor automatizadas pelo sistema e quais tarefas são melhor tratadas manualmente ou por outros sistemas

Sistema - Representação



Sistema - Representação



Atores

❖ Diagrama de casos de uso (atores)

- Um ator é alguém ou algo que interage com o sistema
- É quem ou o que usa o sistema
- Um ator pode ser um departamento, um profissional, uma máquina, etc.
- A interação com o sistema significa que o ator envia ou recebe mensagens para/do sistema, ou troca informações com o sistema
- Para identificar atores, estabelece-se quais objetos estão interessados no uso e interação com o sistema

Atores - Representação

Um ator é representado por um boneco e um rótulo com o nome do ator.

Um ator é um usuário do sistema, que pode ser um usuário humano ou um outro sistema computacional.

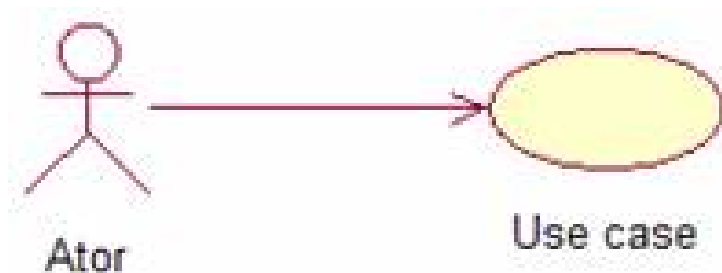
Exemplos: Gerente, Financeiro, Funcion



Relacionamentos (Ator -> Caso de Uso)

Um ator utiliza um caso de uso - relacionamento

- Entre um ator e um *use case*
 - Associação

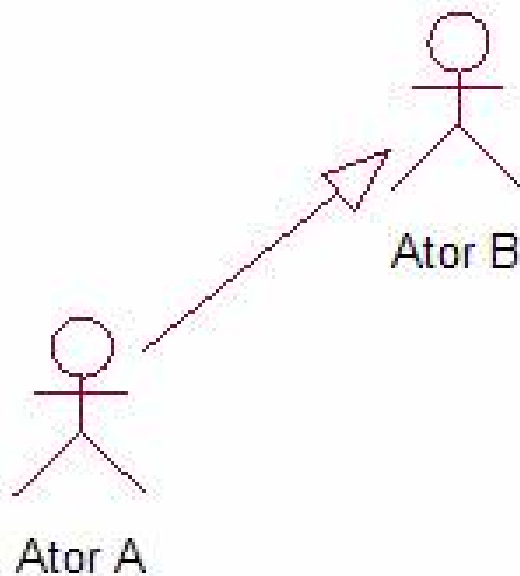


Define uma funcionalidade do sistema do ponto de vista do usuário.

Relacionamentos (Ator -> Outro Ator)

Um ator pode herdar funções executadas por outro ator

- Entre atores
 - Generalização



- Os *use cases* de B são também *use cases* de A

- A tem seus próprios *use cases*

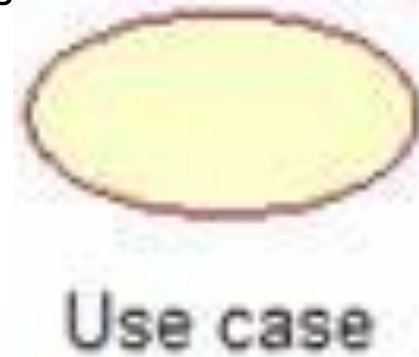
Casos de Uso

- ❖ Um Caso de Uso representa uma funcionalidade do sistema, como percebida por um ator.
- ❖ Define um comportamento do sistema sem revelar a estrutura interna.
- ❖ São os serviços (funcionalidades) que o sistema oferece ou deve oferecer

Casos de Uso - representação

- ❖ Um *use case* é representado por uma elipse e um rótulo com o nome do *use case*.
- ❖ O nome deve usar verbos no infinitivo que facilitem a identificação da funcionalidade do mesmo.

Exemplos: Controlar ..., Manter ..., Registrar ..., Cadastrar ..., Aprovar ...

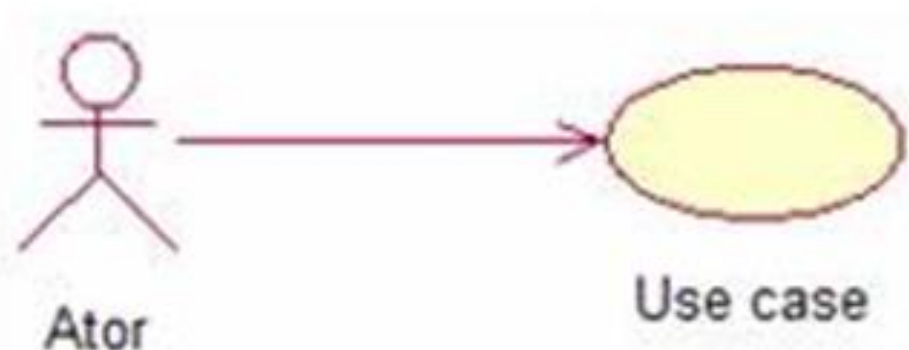


Relacionamento

- ❖ Um ator deve estar relacionado a um ou mais casos de uso.
- ❖ Os casos de uso podem se relacionar.
- ❖ Tipo de relacionamento
 - Comunicação
 - Extensão
 - Inclusão
 - Generalização

Relacionamento de Comunicação

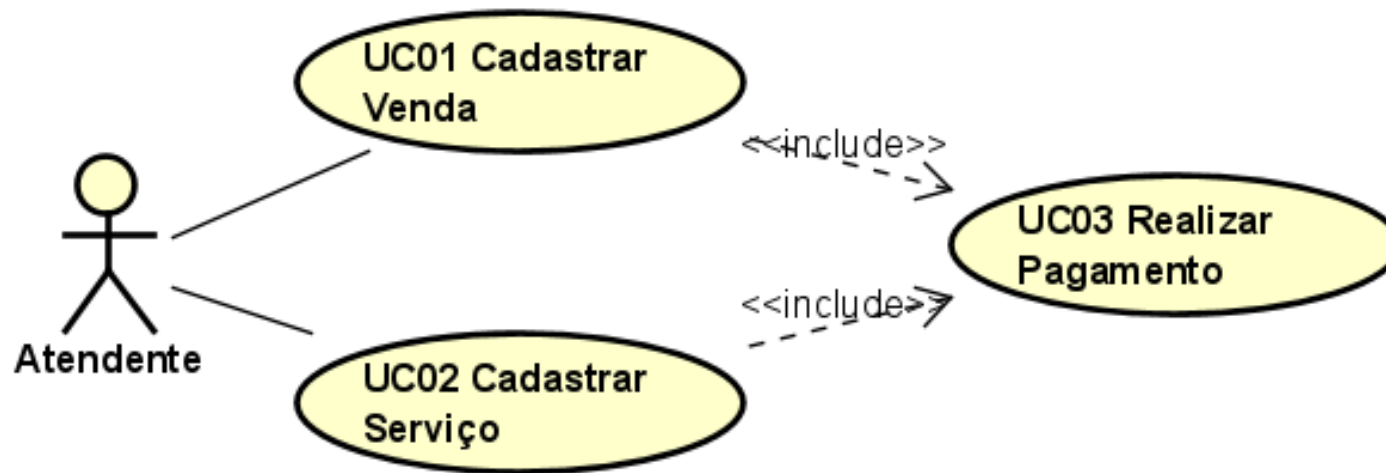
- ❖ Quais atores estão relacionados a que casos de uso.
- ❖ Um ator pode se relacionar com mais de um caso de uso do sistema.
- ❖ É o mais comumente usado de todos os relacionamentos.



Relacionamento por Inclusão

- ❖ Existe somente entre casos de uso.
- ❖ Quando dois ou mais casos de uso incluem uma sequência comum de interações, essa sequência comum pode ser descrita em um outro caso de uso. A partir daí, os casos de uso do sistema podem usar esse caso de uso comum.

Relacionamento por Inclusão - Exemplo

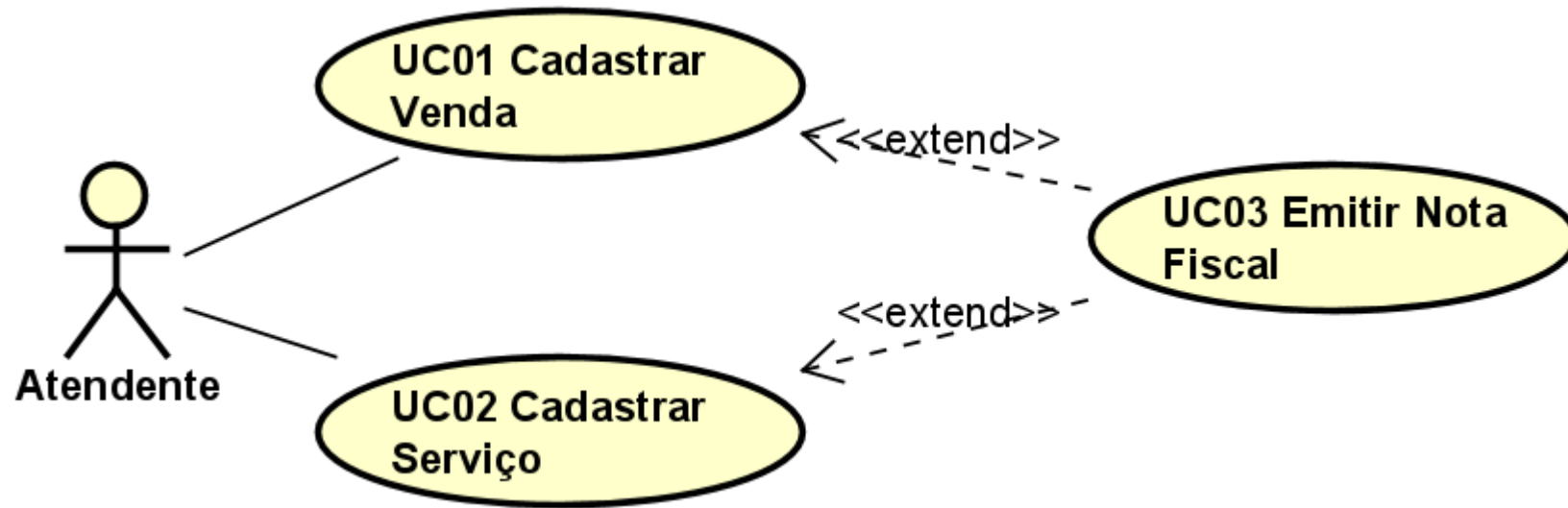


- ❖ Utilizado para demonstrar que um caso de uso pode ser utilizado por vários outros casos de uso, quando este representa uma atividade comum a mais casos de uso
- ❖ **Apresenta uma relação de dependência**, ou seja, se Cadastrar Venda e/ou Cadastrar Serviço forem executados, Realizar Pagamento obrigatoriamente será executado
- ❖ O caso de uso **Realizar pagamento** pode ser chamado por **Cadastrar venda** ou por **Cadastrar serviço**

Relacionamento por Extensão

- ❖ Existe somente entre casos de uso.
- ❖ É utilizado para modelar situações em que diferentes sequências de interações podem ser inseridas (opcionalmente) em um caso de uso (estendido).
 - Comportamento só ocorre sob certas condições.
 - Ou sua realização depende da escolha de um ator.

Relacionamento por Extensão - Exemplo



- ❖ O desenho informa que a emissão da nota pode ser chamada diretamente do cadastro da venda ou cadastrar serviço, mas esta é uma ação que **pode ocorrer ou não**
- ❖ O caso de uso que estende tem uma relação de dependência com o caso de uso estendido (seta tracejada), ou seja, o **Emitir nota** só pode ser executado se **Cadastrar venda** for executado antes

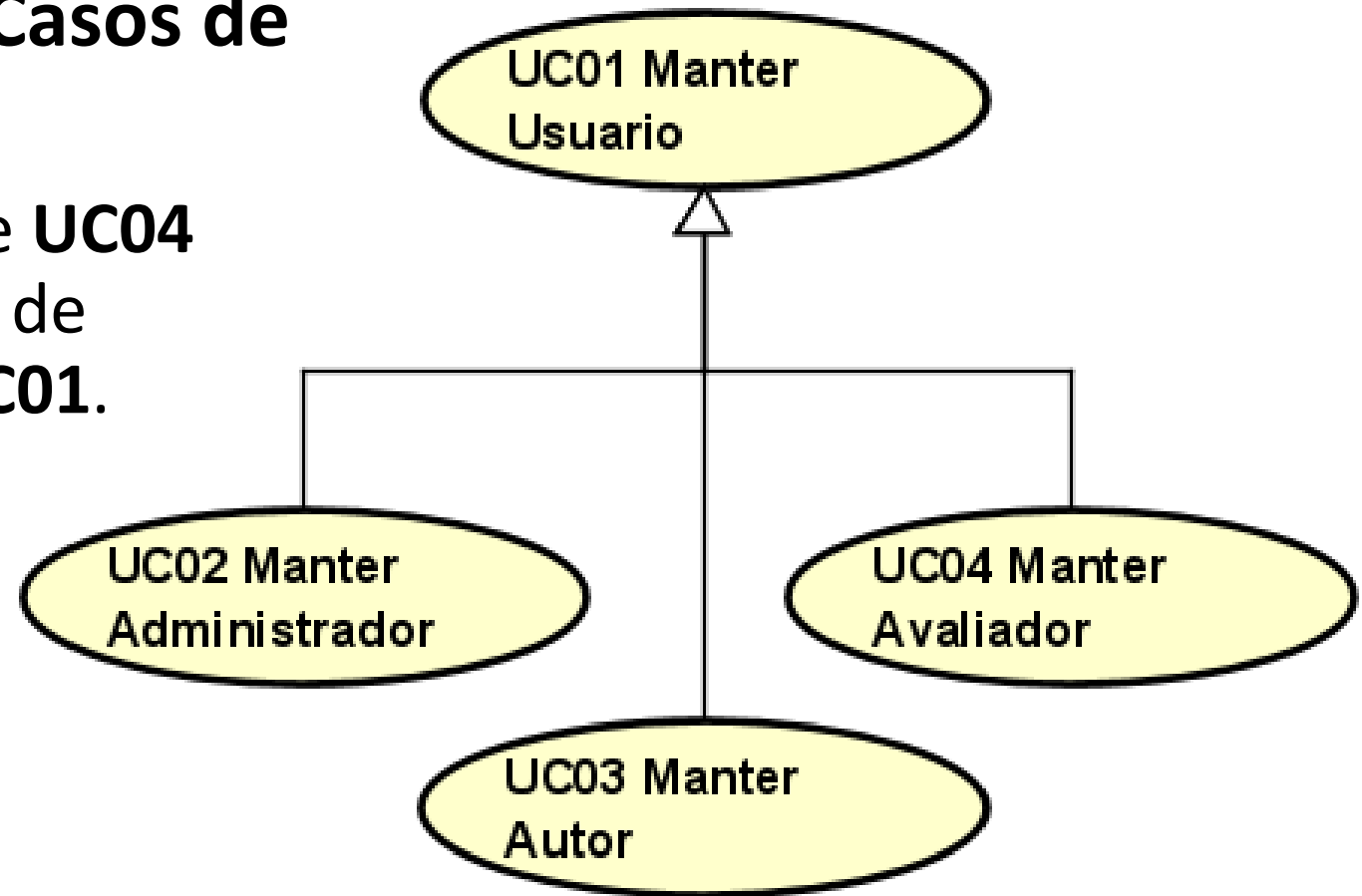
Relacionamento por Generalização

- ❖ Pode existir entre dois casos de uso ou entre dois atores.
- ❖ Permite que um caso de uso (ou um ator) herde características de um caso de uso (ou ator) mais genérico.

Relacionamento por Generalização

❖ Generalização entre Casos de Uso

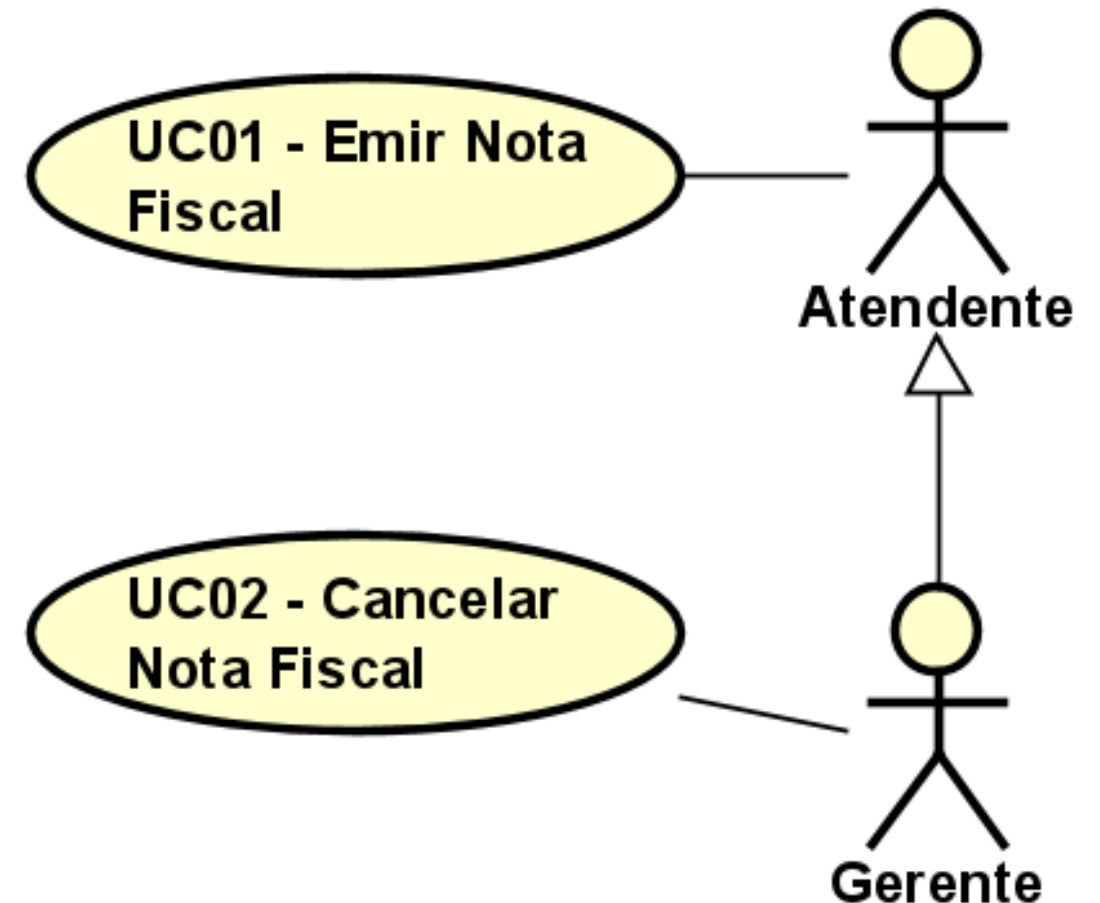
- Quando **UC02**, **UC03** e **UC04** herdam as sequências de comportamento de **UC01**.



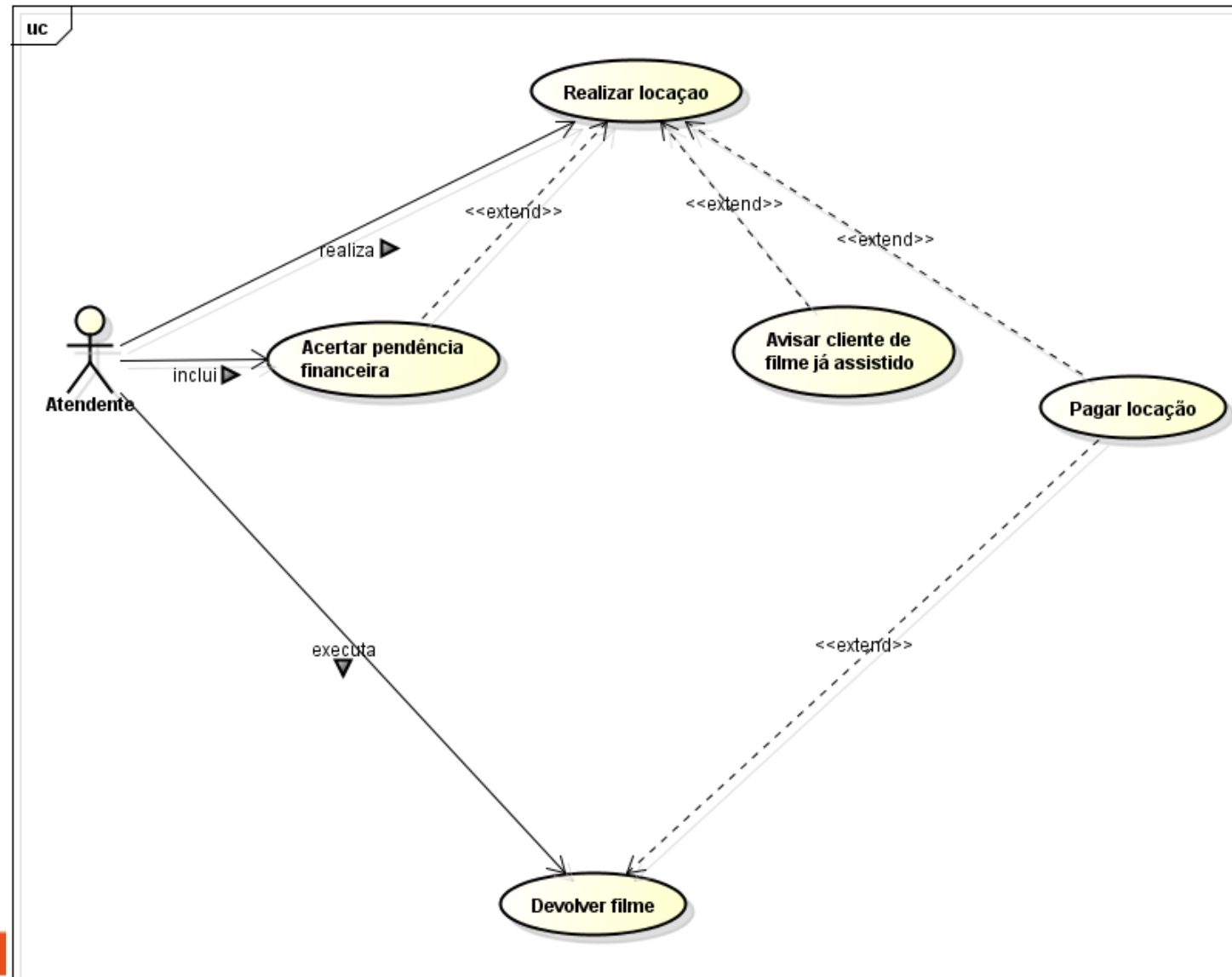
Relacionamento por Generalização

❖ Generalização entre Atores

- O ator herdeiro possui o mesmo comportamento (em relação ao sistema) que o ator do qual ele herda.
- O ator herdeiro pode participar em casos de uso em que o ator do qual ele herda não participa.



Exemplo de Diagrama de Casos de Uso



Dicas para identificar atores

- ❖ Quem utilizará a principal funcionalidade do sistema (atores principais)?
- ❖ Quem necessita suporte do sistema para realizar suas tarefas diariamente?
- ❖ Quem irá manter, administrar e fazer com que o sistema permaneça operando (atores coadjuvantes)?

Dicas para identificar atores

- ❖ Quem ou o quê tem interesse nos resultados produzidos pelo sistema?
- ❖ Quem proverá suporte ao sistema em seu processamento diário?
- ❖ Quais dispositivos de hardware são necessários ao sistema?
- ❖ Com quais outros sistemas o sistema em foco irá interagir?

Dicas para identificar casos de uso

- ❖ O ator precisa ler, criar, destruir, modificar ou armazenar algum tipo de informação no sistema? (verbo!)
- ❖ O quê o sistema necessita de entrada e de saída?
- ❖ Quais as funções que o ator necessita do sistema?
- ❖ O que o ator necessita fazer?

Dicas para identificar casos de uso

- ❖ Quais são as entradas e as saídas, juntamente com sua origem e destino, que o sistema requer?
- ❖ O ator tem de ser notificado sobre eventos no sistema ou ainda notificar o sistema quando algo ocorre?
- ❖ O trabalho diário do ator pode ser simplificado ou tornado mais eficiente através de novas funções do sistema?
- ❖ Quais são os principais problemas com a implementação atual do sistema?

Dicas interessantes para construção do diagrama

- ❖ Melhor fazer menos do que fazer demais.
- ❖ Breve e fácil de ler.
- ❖ Preferência na descrição textual.
- ❖ Limitar os relacionamentos com **<<include>>** e **<<extend>>**.

O que não colocar no diagrama de UC

- ❖ Textos longos.
- ❖ Muitas extensões.
- ❖ Todos diagramas se chamando.
- ❖ Todas as ações CRUD separadas, salvo alguma exceção.
- ❖ Detalhes da tela (botões, combos, links, etc).
- ❖ Não é um fluxograma.

Atenção: Não relacione o caso de uso com as classes do sistema.

Bibliografia

- ❖ DEBONI, J. E. Z. *Modelagem Orientada a Objetos com a UML*. 1ª ed. Futura, 2003.
- ❖ FURLAN, J. D. *Modelagem de Objetos Através da UML: The Unified Modeling Language*. 1ª ed. Makron Books, 1998.
- ❖ GUEDES, G. T. A. *UML – Uma Abordagem Prática*. 2ª ed. Novatec, 2006.
- ❖ MEDEIROS, E. S. *Desenvolvendo Software com UML 2.0 – Definitivo*. 1ª ed. Makron Books, 2004.
- ❖ MELO, A. C. *Desenvolvendo Aplicações com UML 2.0 – Do Conceitual à Implementação*. 2ª ed. Brasport, 2004.
- ❖ PRESSMAN, R. *Engenharia de Software*. 5ª ed. McGraw-Hill, 2002.
- ❖ SILVA, R. P. *UML 2 em Modelagem Orientada a Objetos*. 1ª ed. Visual Books, 2007.