
Slides 10 - requisitos e diagrama de Casos de Uso

— Sebastião Emidio Alves Filho —

Introdução

- Caso de uso é um conceito fundamental de muitas metodologias de desenvolvimento orientada a objetos
- Eles expressam as expectativas de clientes / stakeholders (interessados) no sistema
 - Essencial para um projeto detalhado
- Em geral, o diagrama de Casos de Uso é usado durante todo o processo de análise e projeto
- Pode-se usar o diagrama de Casos de Uso para responder às seguintes questões de projeto:
 - O que está sendo descrito? (O sistema)
 - Quem interage com o sistema? (Os atores)
 - O que os atores fazem (Os casos de uso)

Exemplo: Sistema de gestão acadêmica

■ Sistema

(O que está sendo descrito?)

- O sistema de gestão acadêmica

■ Atores

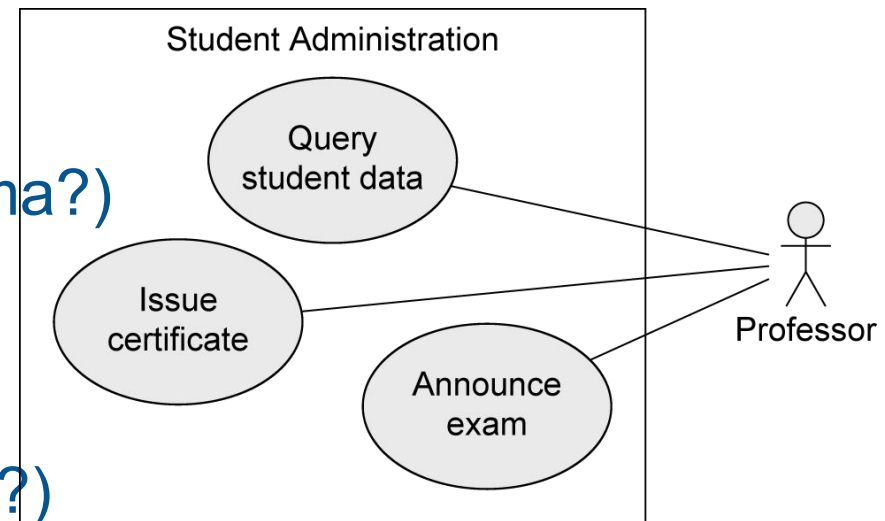
(Quem interage com o sistema?)

- Professor

■ Casos de uso

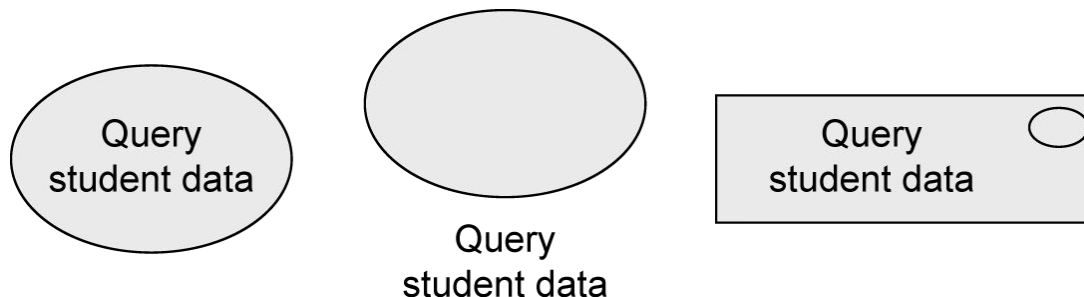
(O que os atores podem fazer?)

- Consultar dados dos estudantes
- Emitir certificados
- Comunicar data de prova



Caso de uso

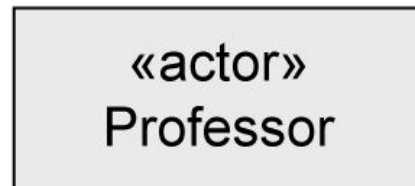
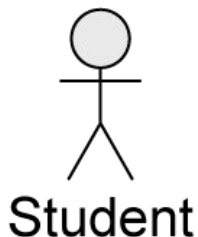
- Descreve uma funcionalidade esperada / implementada de um sistema em construção ou já construído
- Propicia um benefício para um ou mais atores que se comunicam com este caso de uso
- É derivado de desejos / necessidades colhidas junto ao cliente
- O conjunto de todos os casos de uso descreve as funcionalidades que o sistema deve prover
 - Documenta a funcionalidade que o sistema oferece.
- Formas de representação (notações):





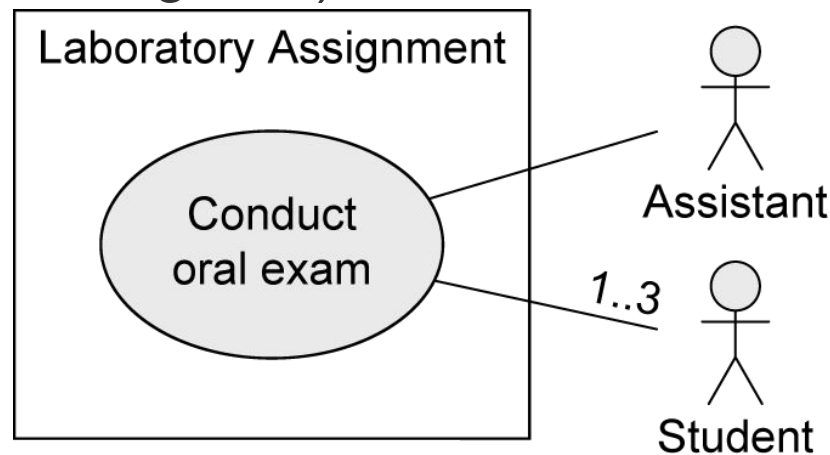
Atores (1/3)

- Atores interagem com o sistema:
 - Usando os casos de uso, isto é, os atores iniciam a sua execução
 - Sendo usados pelos casos de uso, isto é, os atores proveem alguma funcionalidade para a execução dos casos de uso
- Atores representam papéis que usuários e outros sistemas assumem
 - Alguns usuários podem assumir e executar múltiplos papéis simultaneamente
- Atores não são parte do sistema, ou seja, eles estão fora dos limites do sistema
- Formas de representação (notações):



Atores (2/3)

- Normalmente os dados dos usuários são administrados dentro do sistema na forma de classes e objetos
- Exemplo: ator **Tutor**
 - O ator **Tutor** interage com o sistema **Ambiente de Avaliação** ao utilizá-lo.
 - Uma classe Tutor dentro do sistema representa os dados do usuário (nome, login, ...).



Atores (3/3)

- Os atores podem ser:

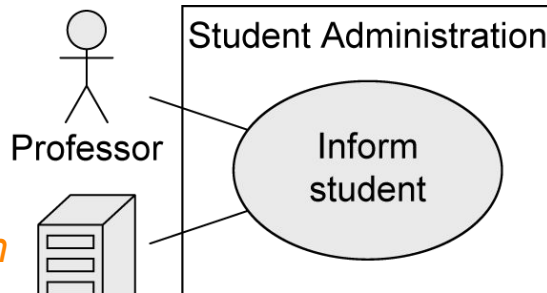
- **Humano:** aluno, professor,...
- **Não-humano:** Serviço de e-mail, Sistema de segurança

- **Primário:** principal beneficiário do caso de uso
- **Secundário:** não recebe o benefício diretamente

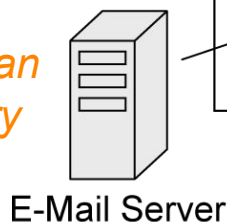
- **Ativo:** inicia a execução do caso de uso
- **Passivo:** provê alguma funcionalidade para a execução do caso de uso

- Exemplo:

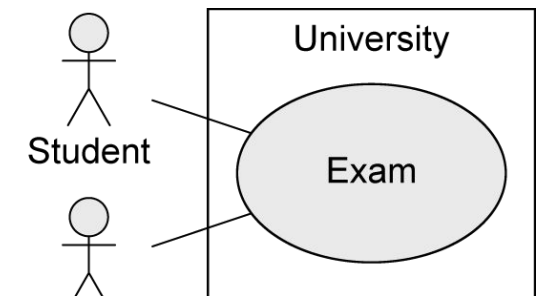
*Human
Primary
Active*



*Non-human
Secondary
Passive*



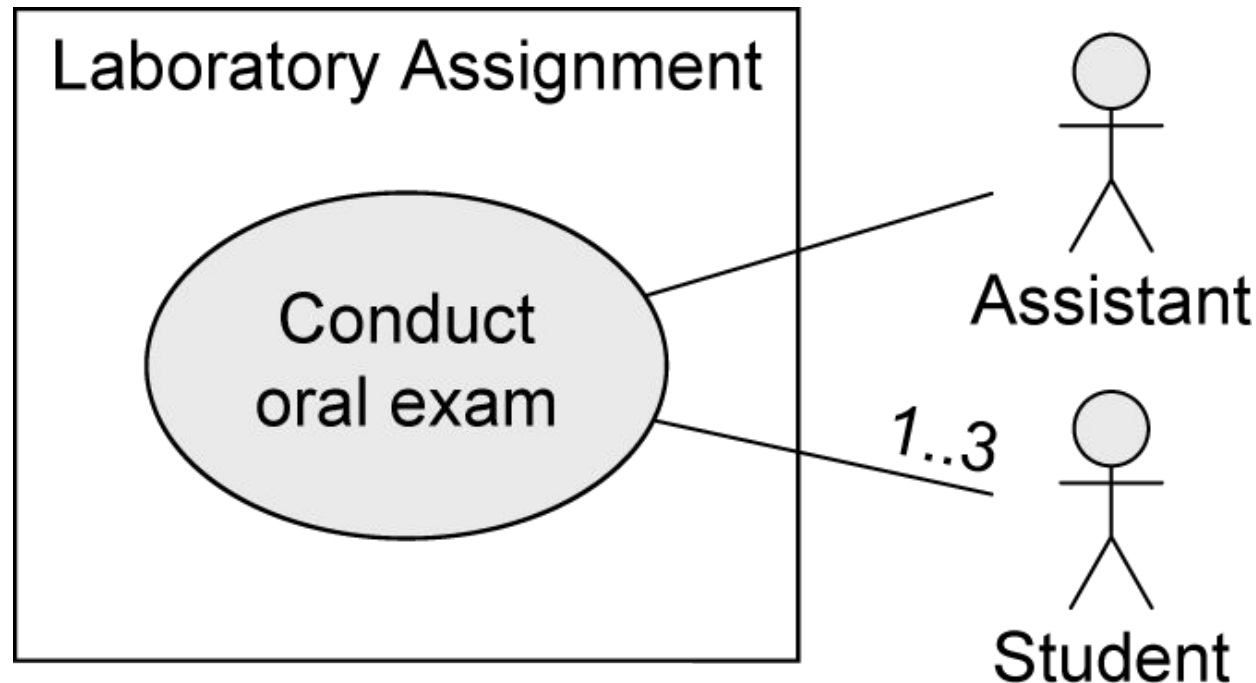
*Human
Primary
Active*



*Human
Secondary
Active*

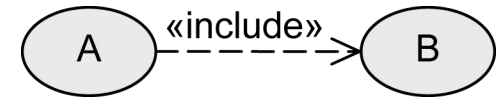
Relacionamento entre caso de uso e atores

- Atores são conectados com os casos de uso através das **Associações** (linhas sólidas)
- Cada ator deve se comunicar com pelo menos um caso de uso
- Uma associação é sempre binária (liga um ator a um caso de uso)
- Pode-se especificar multiplicidades

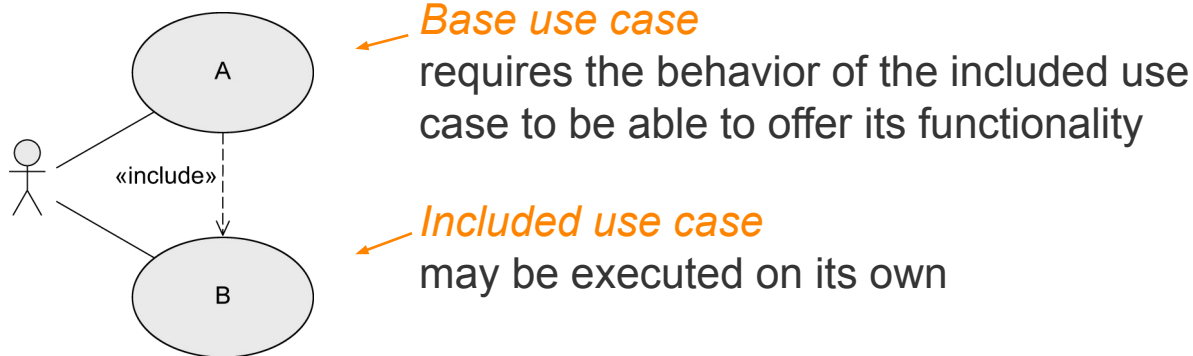


Relações entre casos de uso

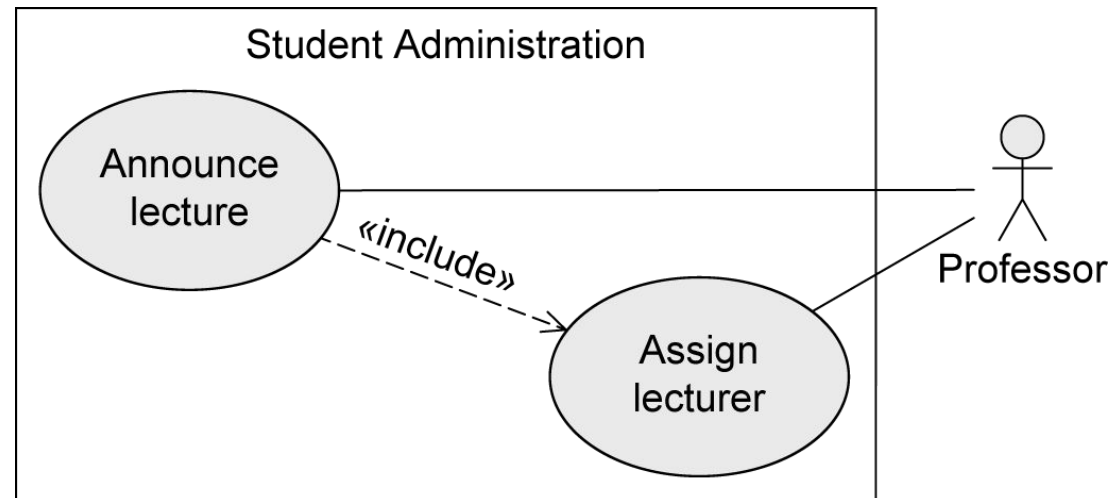
«include»



- O comportamento de um caso de uso (incluso) é integrado no comportamento de outro caso de uso (base)

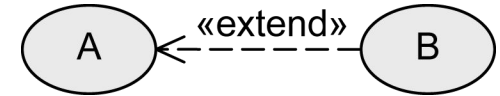


- Exemplo:

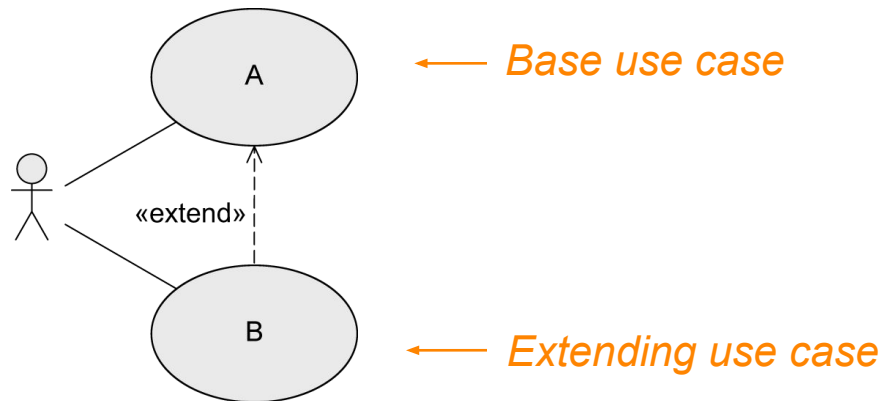


Relacionamento entre casos de uso

«extend»



- O comportamento de um caso de uso (estendido) pode ser integrado no comportamento de outro caso de uso (base) mas não é obrigatório
- Ambos podem ser executados independentemente;

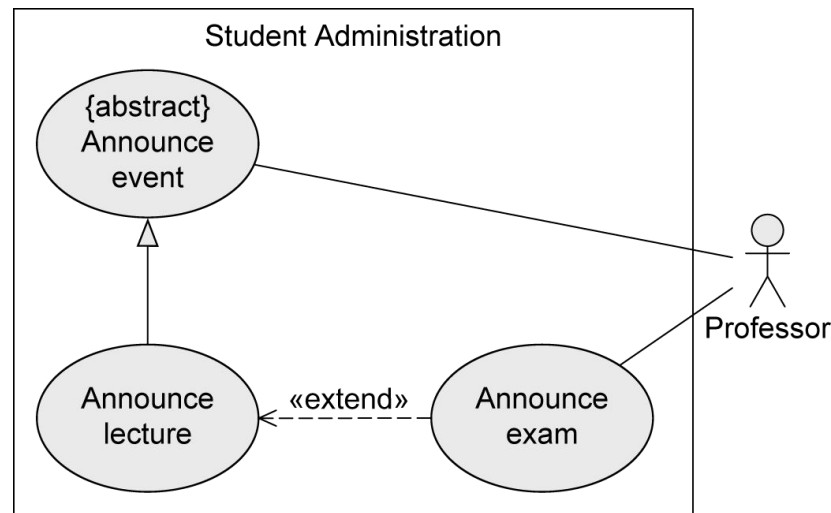
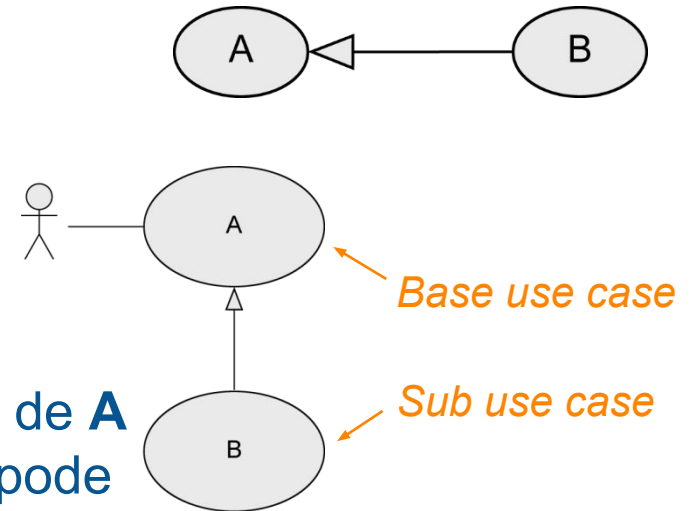


- O caso de uso base A decide se incluirá ou não o caso estendido B
- **Pontos de extensão** definem em que ponto o comportamento é integrado enquanto **condições** definem sob que circunstâncias isso ocorre

Relacionamento entre casos de uso

Generalização

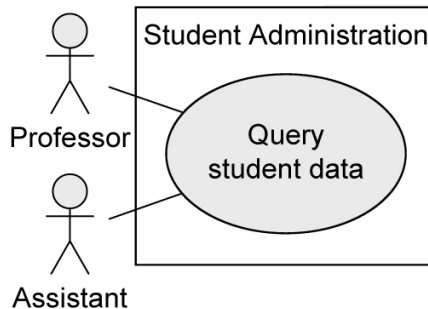
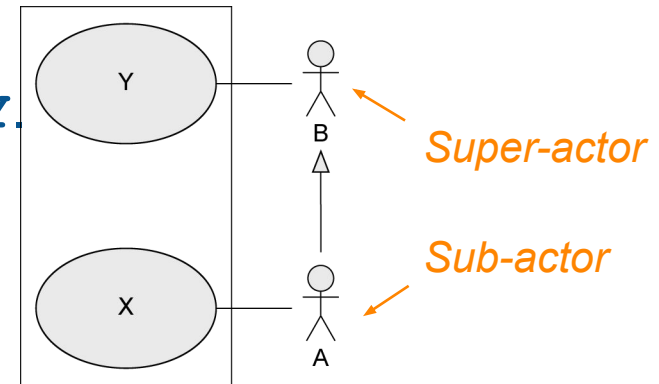
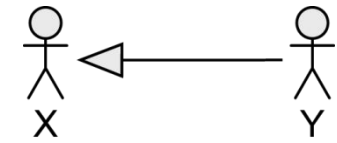
- Caso de uso **A** generaliza caso de uso **B**
- **B** herda o comportamento de **A**, podendo estendê-lo ou sobrescrevê-lo
- **B** também herda todos os relacionamentos de **A**
- **B** adota o funcionamento básico de **A** mas pode decidir que parte de **A** é mantida ou modificada
- **A** pode ser rotulada como **abstrata**
 - Não é executada diretamente
 - Somente **B** é executável
- Exemplo:



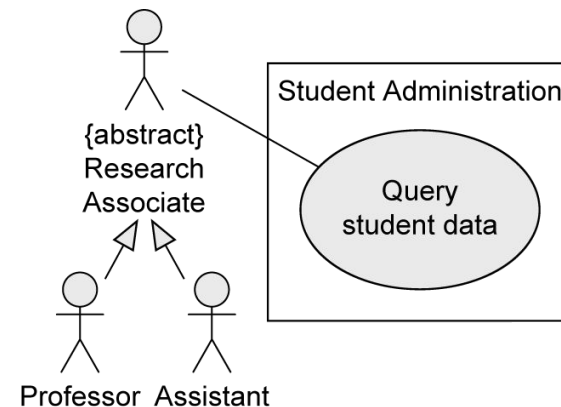
Relacionamento entre atores

Generalização de atores

- Ator **A** herda do ator **B**.
 - **A** se comunica com os casos de uso **X** e **Y**.
 - **B** se comunica somente com **Y**.
 - *Herança múltipla é permitida*
 - É possível criar atores *abstratos*
-
- Exemplos:



≠



Professor AND Assistant
needed
for executing **Query student data**

Professor OR Assistant
needed
for executing **Query student data**

Descrição de casos de uso

■ Abordagem estruturada

- Nome
- Descrição curta
- Pré-condição: pré-requisitos necessários para a execução
- Pós-condição: estado do sistema depois da execução
- Situações de erro: erros relevantes para o domínio do problema e o estado do sistema na ocorrência de um erro
- Atores que se comunicam com o caso de uso
- Trigger(gatilho): evento que origina/inicia o caso de uso
- Processo padrão: passos individuais a serem executados
- Processos alternativos: outros caminhos além do processo padrão

[A. Cockburn: Writing Effective Use Cases, Addison Wesley, 2000]

Exemplo de descrição de caso de uso

- Nome: Reservar auditório.
- Descrição curta: um funcionário reserva um auditório da universidade para um evento.
- Pré-condição: o funcionário tem autorização para fazer a reserva.
- Pós-condição: o auditório está reservado.
- Situação de erro: o auditório não está livre e a reserva não é realizada.
- Ator: **Funcionário**.
- Gatilho: Funcionário faz uma requisição para reserva.
- Processo padrão:
 1. Funcionário loga no sistema;
 2. Funcionário seleciona o auditório desejado;
 3. Funcionário seleciona a data e horário;
 4. Sistema confirma que o auditório está livre;
 5. Funcionário confirma a reserva.
- Processos alternativos
 4. Sistema informa que o auditório não está livre;
 5. Sistema mostra outros auditórios disponíveis no dia e horário pretendido
 6. Funcionário seleciona outro auditório alternativo e confirma a reserva.

Boas práticas

Identificação de atores

- Quem usa os principais casos de uso?
- Quem precisa de suporte para o trabalho diário com o sistema?
- Quem é responsável pela administração do sistema?
- Quais são os sistemas (dispositivos ou softwares) com os quais o sistema modelado se comunica?
- Quem está interessado nos resultados produzidos pelo sistema?

Boas práticas

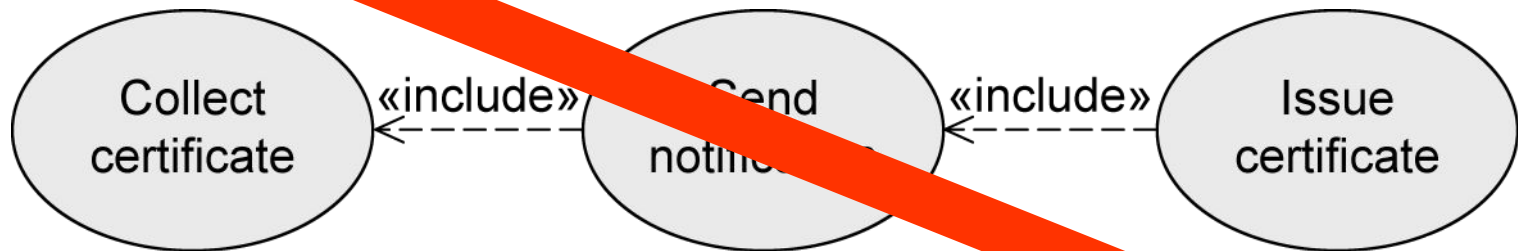
Identificação de Casos de Uso

- Quais as principais tarefas que um Ator deve realizar?
- Um ator deseja consultar ou modificar uma informação contida no sistema?
- Um ator deseja informar ao sistema sobre mudanças em outros sistemas?
- Um ator deve ser informado sobre eventos inesperados ocorridos no sistema?

Boas práticas

Erros a serem evitados(1/5)

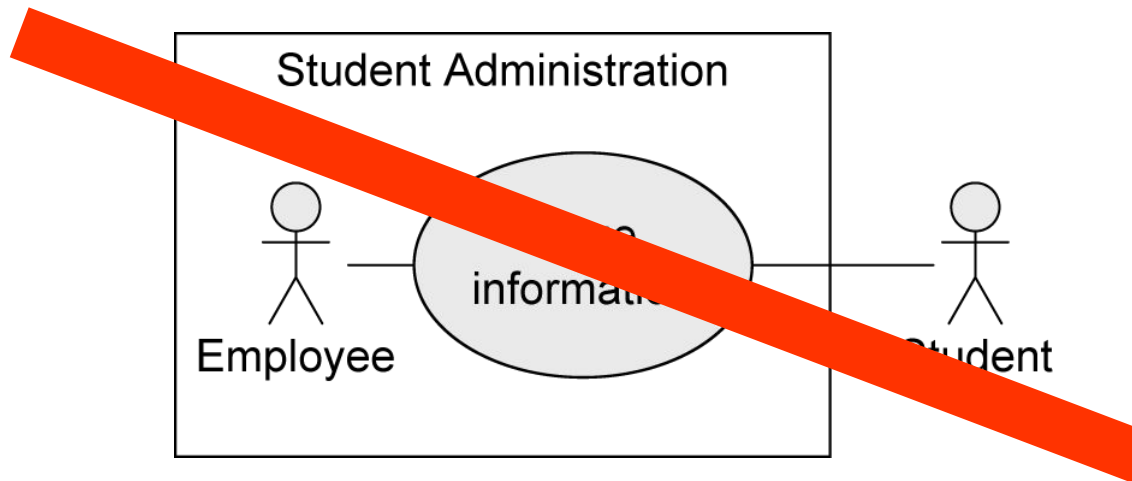
- Fazer diagramas de caso de uso para modelar um processo ou um fluxo de execução de trabalho (workflow)



Boas práticas

Erros a serem evitados(2/5)

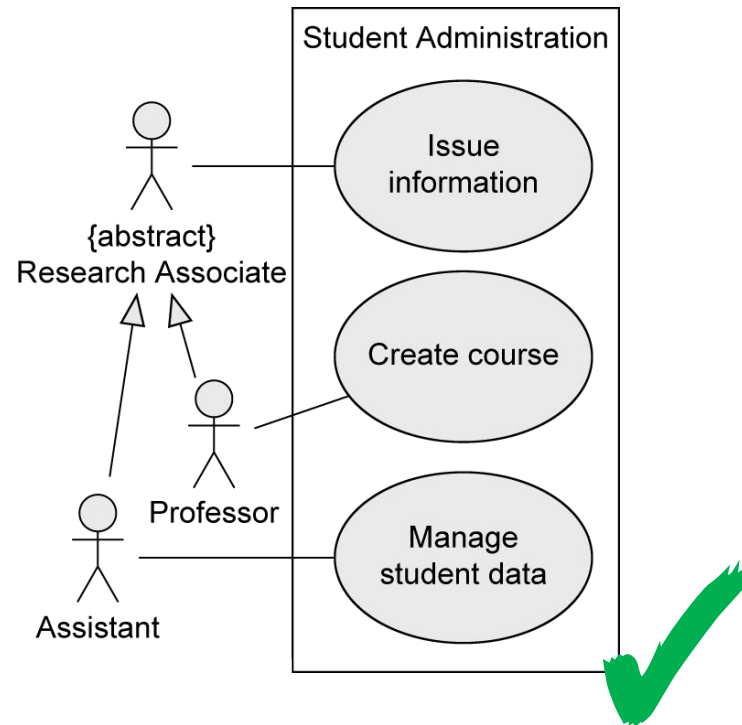
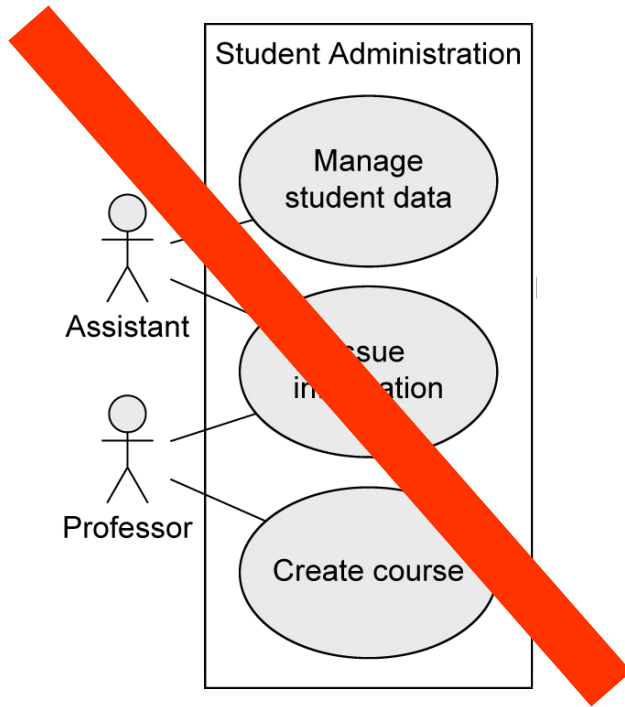
- Atores são elementos externos ao sistema, e não parte dele. Portanto, eles devem ficar fora dos limites do sistema.



Boas práticas

Erros a serem evitados(3/5)

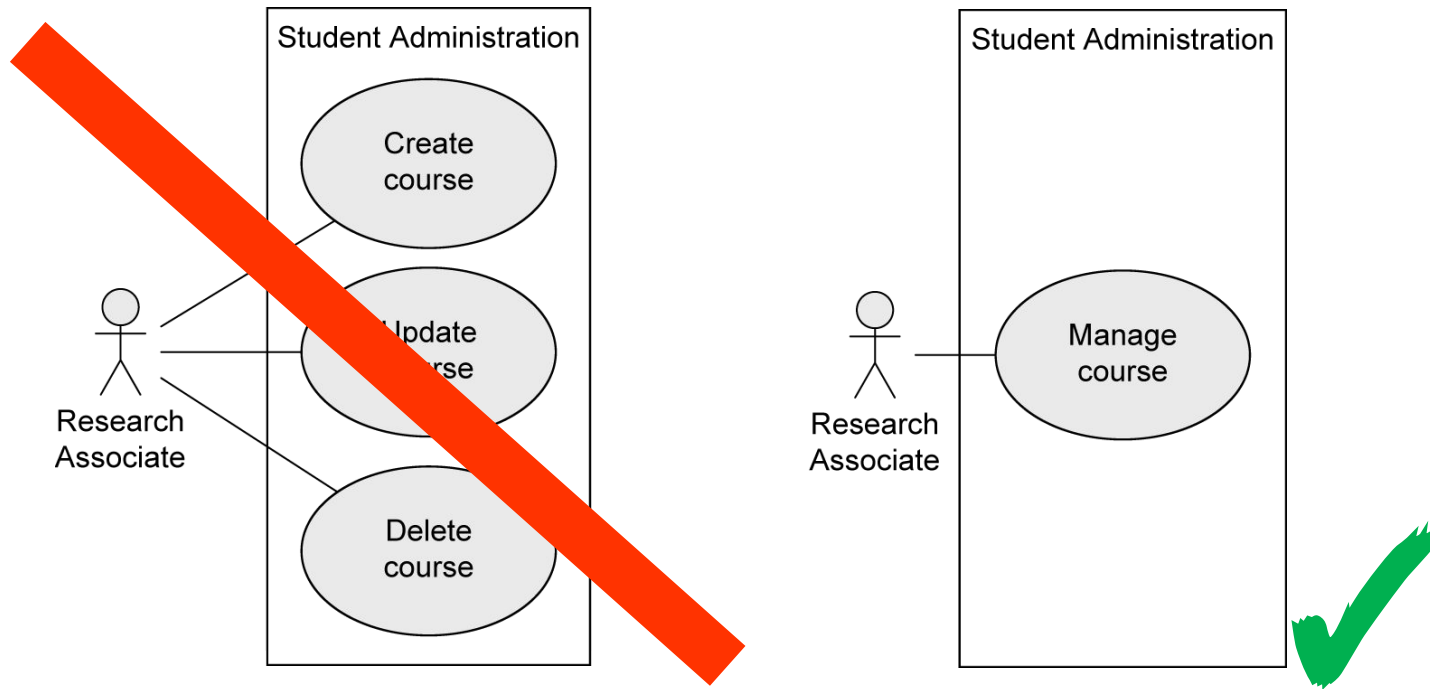
- Quando é necessário apenas um ator para a execução do Caso de Uso e há a possibilidade de realizá-lo com diferentes atores



Boas práticas

Erros a serem evitados (4/5)

- Usar vários pequenos casos de uso para ações que podem ser agrupadas em um único caso.



Boas práticas

Erros a serem evitados (5/5)

- Os passos para executar um caso de uso não devem figurar como outros casos de uso a serem incluídos. Não fazer uma decomposição funcional.

