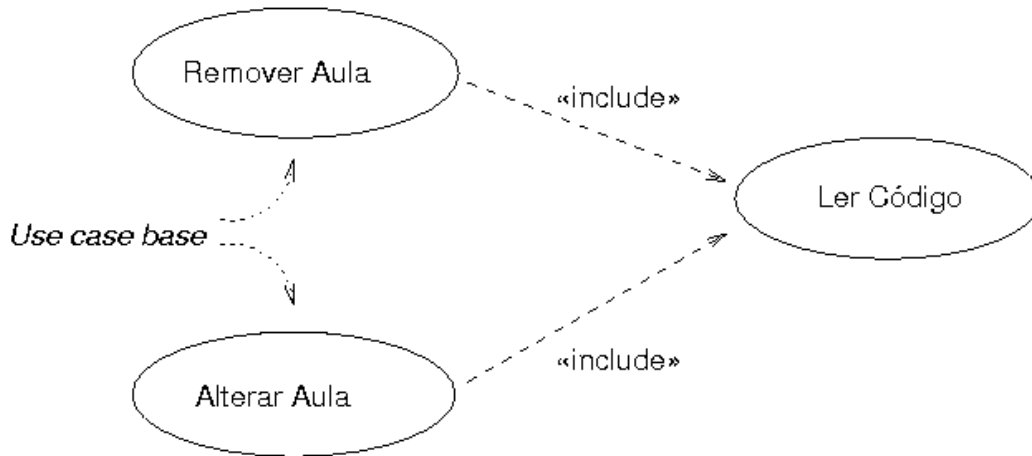


Diagramas de *Use Cases* - conceitos avançados

«include»

- Um estereótipo de dependência.
- Utilizado para indicar a reutilização de comportamento.



- Actores utilizam os *use case base*.
- Quando o *use case base* é executado, também o *use case* incluído o é

Use Case: *Remover Aula*

Comportamento normal:

1. «include» Ler código
2. Sistema apresenta Aula
3. Actor confirma remoção
4. Sistema confirma realização da remoção

Comportamento Alternativo:

3. Actor cancela remoção

Comportamento Alternativo:

2. Sistema indica que não existe aula com esse código.

Use Case: *Alterar Aula*

Comportamento normal:

1. «include» Ler código
2. Sistema apresenta Aula
3. Actor altera dados e confirma alteração
4. Sistema confirma realização da alteração

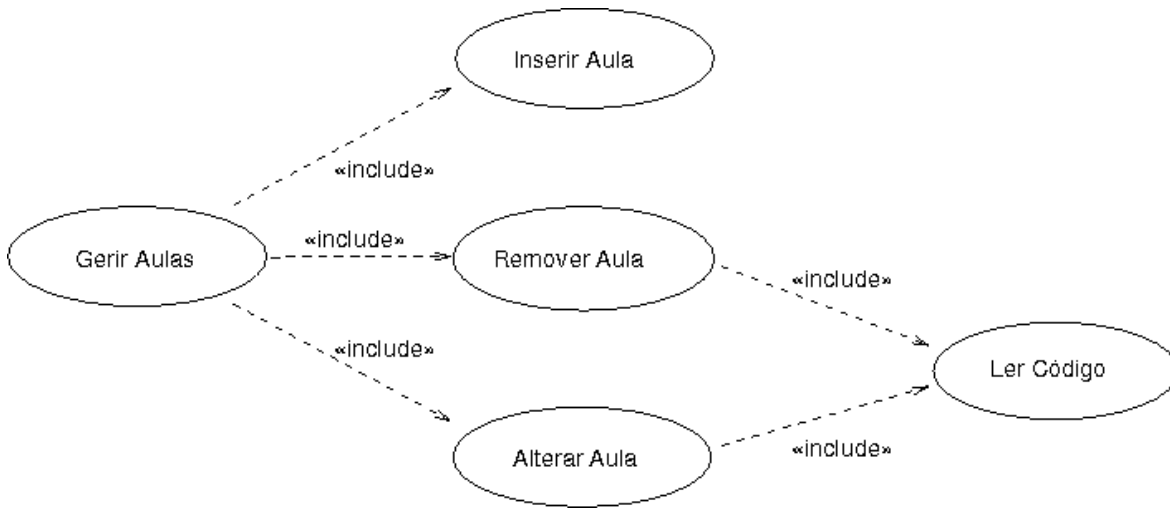
Comportamento Alternativo:

3. Actor cancela alteração

Comportamento Alternativo:

2. Sistema indica que não existe aula com esse código.

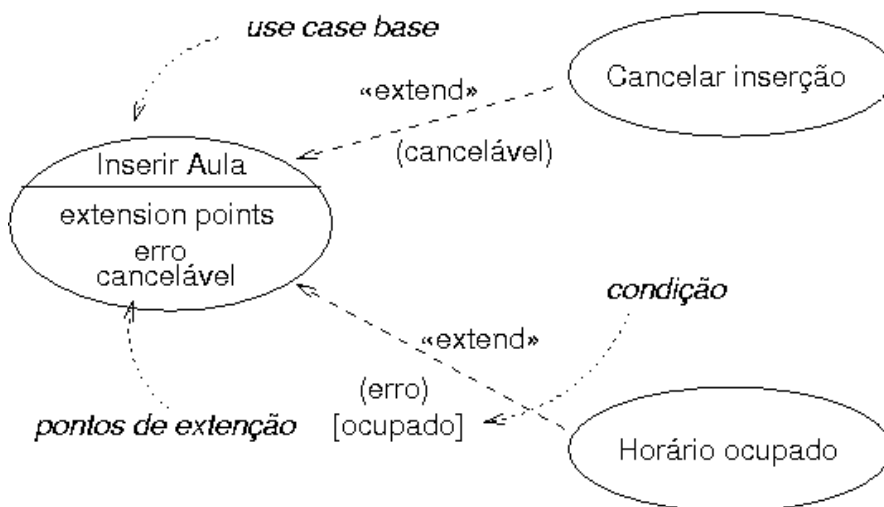
- Também pode ser utilizado para estruturar *use cases*.



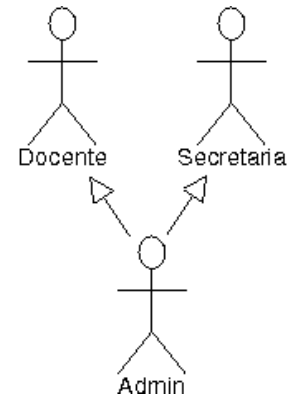
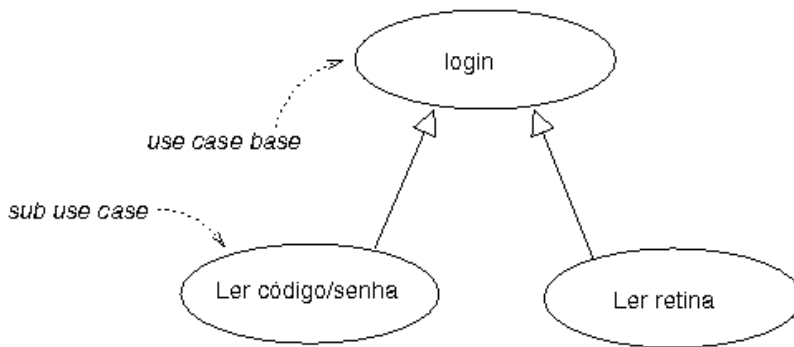
- Não exagerar!
- Em alternativa, utilizar sub-diagramas.

«*extend*»

- Outro estereótipo de dependência.
- Permite adicionar comportamento a um *use case* base.



- Estratégia: escrever caso base; identificar variações; utilizar extensões para elas.
- Caso base deve ser um *use case* bem formado sem as extensões!
- Extensão pode não ser um *use case* bem formado por si só.



- Sub-elementos são casos particulares de super-elementos.
- Um sub-elemento pode ser utilizado onde quer que o super-elemento possa.
- Útil para user profiling (definição de níveis de acesso).
- Nos exemplos apresentados:
 - Existem duas formas de fazer login.
 - O actor Admin pode realizar todos os *use cases* de Docente e Secretaria.

Diagramas de *Use Case* - Resumo

- Os diagramas de *Use Case* permitem definir os requisitos funcionais de um sistema:
 - que serviços deve fornecer;
 - a quem os deve fornecer.
- Notação diagramática facilita o diálogo (com os clientes e dentro da equipa de desenvolvimento).
- Utilizando diagramas de *use case*, clientes e equipa de desenvolvimento podem chegar a um acordo sobre qual o sistema a desenvolver.
- A resolução de alterações nos requisitos funcionais fica facilitada.

No entanto:

- Os diagramas de *use case* não suportam a captura de requisitos não funcionais.

Quando utilizar diagramas de Use Case?

- Sempre que se estiverem a analisar requisitos!



Diagramas de Actividade I

Sumário

- Introdução aos Diagramas de Actividade – Aplicação a *Use Cases*
- Notação base
 - Actividade
 - Estado inicial
 - Estado final
 - Pontos de decisão



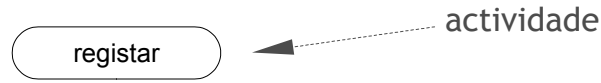
Introdução aos Diagramas de Actividade – Aplicação a *Use Cases*

Descrever um *Use Case*:

- As descrições textuais tendem a ser demasiado longas.
- Os Diagramas de Actividade fornecem uma alternativa para a modelação do comportamento de um dado *Use Case*.
- Tal como nas descrições textuais, a ênfase é colocada nas actividades que ocorrem na interacção entre os actores e o sistema.

Notação base

- Actividade - modela a ocorrência de uma unidade de comportamento (neste caso uma acção por parte de um actor ou por parte do sistema).



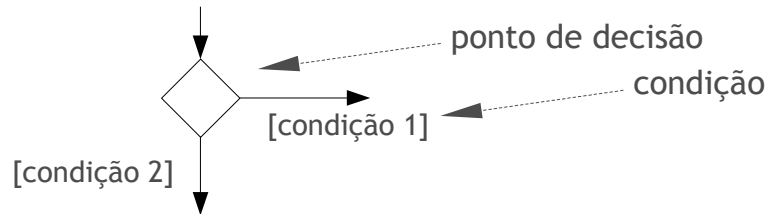
- Estado inicial – início da actividade (neste caso, início do *use case*)



- Estado final – fim da actividade (neste caso, fim do *use case*)



- Ponto de decisão - modela lógica condicional



Diagramas de Estado I

Sumário

- Introdução aos Diagramas de Estado – Aplicação a *Use Cases*
- Notação base
 - Estado
 - Estado inicial
 - Estado final

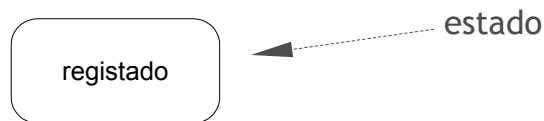
Introdução aos Diagramas de Estado – Aplicação a *Use Cases*

Descrever um *Use Case*:

- As descrições textuais e recorrendo a diagramas de actividade focam os eventos que podem acontecer na interacção entre actor e sistema.
- Os Diagramas de Estado também possibilitam modelar o comportamento de um dado *Use Case*.
- A ênfase é colocada no estado da interface com o utilizador – modelam-se todos os estados possíveis que a interface atravessa em resposta aos eventos gerados pelos actores.

Notação base

- Estado
 - define uma possível estado do objecto (neste caso a interface do sistema);
 - normalmente traduz em valores específicos dos seus atributos.



- Estado inicial – estado do objecto quando é criado



- Estado final – destruição do objecto (neste caso, fim do *use case*)

