



# Desenvolvimento de Sistemas Software

## Modelação do Requisitos Funcionais (Diagramas de *Use Case*)



# Exemplo da aula anterior - Máquina Multibanco

- **Cenários**

...

- **Use Cases**

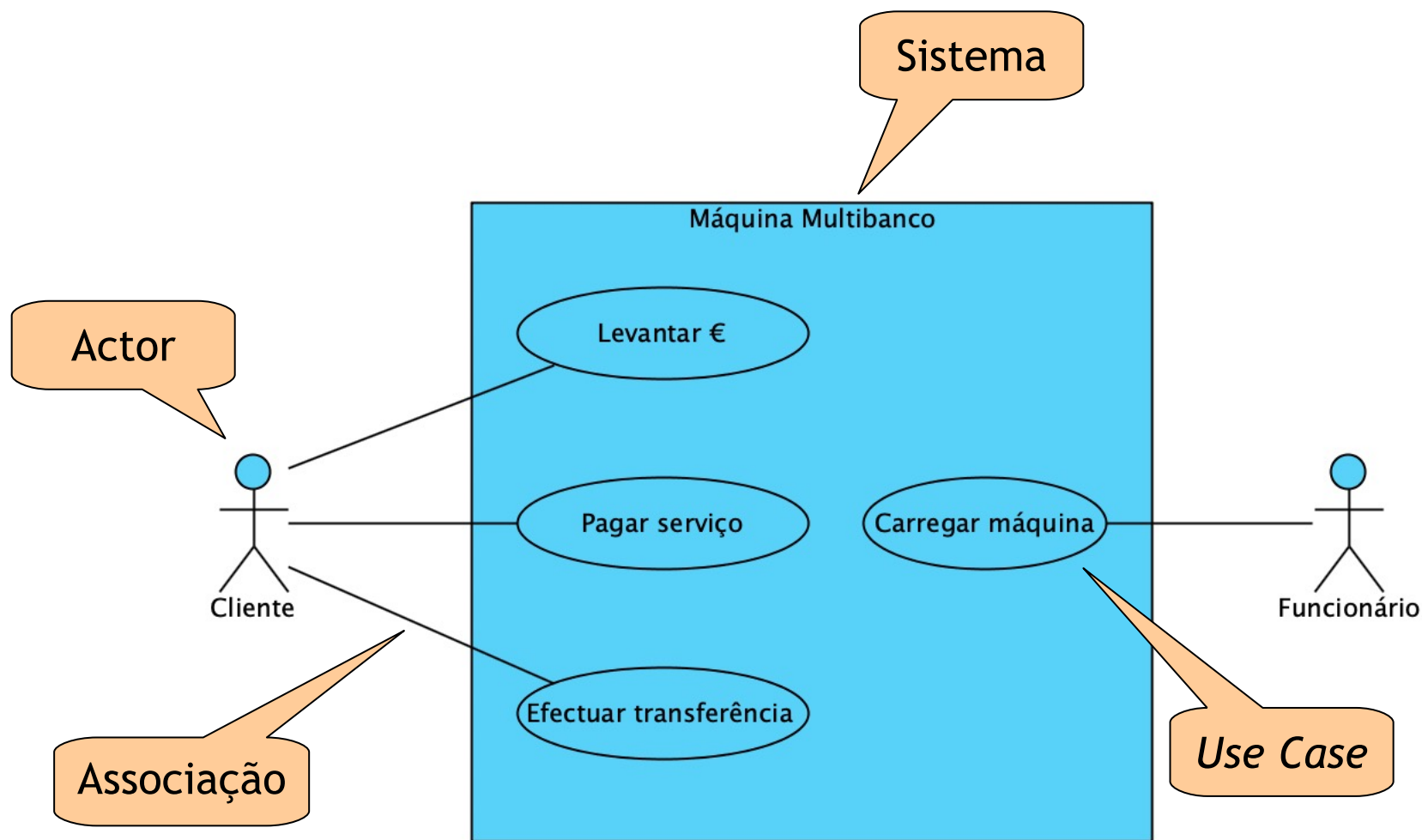
- Levantar €
- Pagar serviço
- Efectuar transferência
- Carregar máquina

- **Actores**

- Cliente
- Funcionário



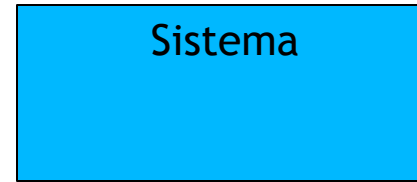
# Diagrama de Use Case - Exemplo





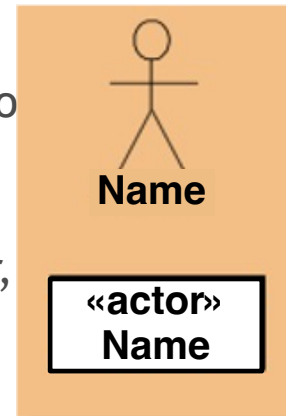
## *Sistema*

- define as fronteiras da solução a desenvolver



## *Actor*

- uma abstracção para uma entidade fora do sistema
- um actor modela um propósito (alguém que tem um interesse específico no sistema) - pode não mapear 1 para 1 com entidades no mundo real
- um actor não é necessariamente um humano - pode ser um computador, outro sistema, etc.
- cada actor representa um papel (“role”) que “alguém” ou qualquer “coisa” externa ao sistema pode assumir
- o conjunto de todos os actores definem todas as formas de interacção com o sistema



## *Associação*

- representa comunicação entre o actor e o sistema - através de *use cases*
- pode ser bi-direccional ou uni-direccional



# Que Actores? Que Associações?

## *Todas as associações*

- Todos os sistemas externos que interagem com o sistema em análise são apresentados como actores e todas as interacções são representadas nos diagramas.
- Demasiado abrangente, em muitos casos existem interacções com outros sistemas apenas por razões de implementação e não por se tratarem de requisitos do sistema.

## *Apenas as associações relativas a interacção iniciada por sistemas externos*

- Só são representados como actores os sistemas externos que iniciem diálogo com o sistema em análise.
- Mesmo assim muito abrangente.



# Que Actores? Que Associações?

## ***Apenas as associações em que é o sistema externo o interessado***

- Neste caso só são apresentados como actores os sistemas externos que necessitam de funcionalidade fornecida pelo sistema em análise.
- Usalmente esta é uma solução equilibrada.

## ***Não mostrar associações com sistemas externos***

- Apenas os utilizadores são actores, neste caso quando existem sistemas externos apresentam-se os seus actores em diálogo directo com o sistema a ser modelado.
- De uma outra forma esta solução também é demasiado abrangente e pode levar a confusão sobre quem está realmente a utilizar o sistema.



## Use Cases - ponto situação

- Forma sistemática de capturar requisitos funcionais
  - que serviços deve fornecer; a quem os deve fornecer
  - mas **não** suportam a captura de requisitos não funcionais
- Notação diagramática facilita o diálogo
  - com os clientes e dentro da equipa de desenvolvimento
- Modelam o contexto geral do sistema
  - Quais os actores que com ele se relacionam
- Especificam todas as possíveis utilizações
  - O que cada actor pode fazer no sistema
- O centro de todo o processo de desenvolvimento
  - Desde a concepção da arquitectura, passando pelos testes, até ao manual de utilização...



## *Use Cases - ponto situação*

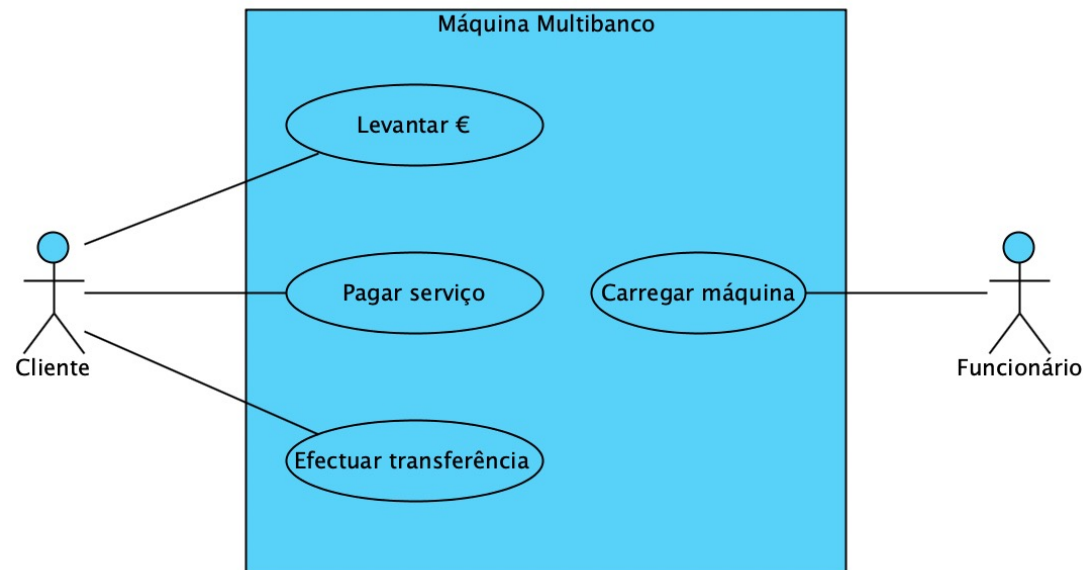
- A concepção do sistema é guiada pelo modelo de *Use Case*:
  - Utilizando diagramas de *Use Case*, clientes e equipa de desenvolvimento podem chegar a um acordo sobre qual o sistema a desenvolver
- A implementação do sistema é guiada pelo modelo de *Use Case*:
  - cada *Use Case* é implementado sucessivamente:
  - quando todos os *Use Cases* estiverem implementados obtém-se o sistema final;
  - fica facilitada a manutenção sempre que os requisitos sejam alterados;
- O modelo de *Use Case* pode ser utilizado para o planeamento de testes:
  - A partir das definições dos *Use Case*, definir os testes que o Sistema deverá passar.





# Estruturação dos modelos de Use Case

- Dependências entre Use Case (<<include>> / <<extend>>)
- Generalização
- Sub-diagramas
- Exemplo de uma abordagem de *refactoring*...





# Um exemplo...

## Use Case: Levantar €

**Descrição:** Cliente levanta quantia da máquina

**Cenários:** O João levanta €60 com cartão; O João levanta €10 com MB way

**Pré-condição:** Sistema tem notas

**Pós-condição:** Cliente tem quantia desejada e saldo da conta foi actualizado

### **Fluxo normal.**

1. Cliente apresenta cartão e PIN
2. Máquina MB valida acesso e pede operação
3. Cliente indica que pretende levantar dada quantia
4. Máquina MB pergunta se quer talão
5. Cliente responde que não
6. Máquina MB devolve cartão, fornece notas e actualiza saldo
7. Cliente retira cartão e notas

### **Fluxo alternativo (1): [cliente quer talão] (passo 5)**

...

### **Fluxo de excepção (2): [PIN inválido] (passo 2)**

- 2.1. Máquina MB avisa sobre PIN inválido e devolve cartão
- 2.2. Cliente retira cartão

### **Fluxo alternativo (3): [cliente autentica-se com MB way] (passo 1)**

- 1.1. Cliente prime escolhe acesso MB way
- 1.2. Máquina MB pede Código MB way
- 1.3. Cliente indica Código MB way
- 1.4. Regressa a 2



# Um exemplo...

## Use Case: Levantar €

**Descrição:** Cliente levanta quantia da máquina

**Cenários:** O João levanta €60 com cartão; O João levanta €10 com MB way

**Pré-condição:** Sistema tem notas

**Pós-condição:** Cliente tem quantia desejada e saldo da conta foi actualizado

### Fluxo normal:

1. Cliente apresenta cartão e PIN
2. Máquina MB valida acesso e pede operação
3. Cliente indica que pretende levantar dada quantia
4. Máquina MB pergunta se quer talão
5. Cliente responde que não
6. Máquina MB devolve cartão, fornece notas e actualiza saldo
7. Cliente retira cartão e notas

### Fluxo alternativo (1): [cliente quer talão] (passo 5)

### Fluxo de excepção (2): [PIN inválido] (passo 2)

- 2.1. Máquina MB avisa sobre PIN inválido e devolve cartão
- 2.2. Cliente retira cartão

### Fluxo alternativo (3): [cliente autentica-se com MB way] (passo 1)

- 1.1. Cliente prime escolhe acesso MB way
- 1.2. Máquina MB pede Código MB way
- 1.3. Cliente indica Código MB way
- 1.4. Regressa a 2

- Repetição dos mesmos fluxos!
- Acrescentar retenção do cartão após três tentativas?!

## Use Case: Pagar serviço

...

### Fluxo normal:

1. Cliente apresenta cartão e PIN
2. Máquina MB valida acesso e pede operação
3. Cliente indica que pretende pagar um serviço

...

### Fluxo alternativo (1): ...

...

### Fluxo de excepção (2): [PIN inválido] (passo 2)

- 2.1. Máquina MB avisa sobre PIN inválido e devolve cartão
- 2.2. Cliente retira cartão

### Fluxo alternativo (3): [cliente autentica-se com MB way] (passo 1)

- 1.1. Cliente escolhe acesso MB way
- 1.2. Máquina MB pede Código MB way
- 1.3. Cliente indica Código MB way
- 1.4. Regressa a 2

## Use Case: Efectuar transferência

...

### Fluxo normal:

1. Cliente apresenta cartão e PIN
2. Máquina MB valida acesso e pede operação
3. Cliente indica que pretende realizar uma transferência

...

### Fluxo alternativo (1): ...

...

### Fluxo de excepção (2): [PIN inválido] (passo 2)

- 2.1. Máquina MB avisa sobre PIN inválido e devolve cartão
- 2.2. Cliente retira cartão

### Fluxo alternativo (3): [cliente autentica-se com MB way] (passo 1)

- 1.1. Cliente prime escolhe acesso MB way
- 1.2. Máquina MB pede Código MB way
- 1.3. Cliente indica Código MB way
- 1.4. Regressa a 2



# Um exemplo...

## Use Case: Levantar €

**Descrição:** Cliente levanta quantia da máquina

**Cenários:** O João levanta €60 com cartão; O João levanta €10 com MB way

**Pré-condição:** Sistema tem notas

**Pós-condição:** Cliente tem quantia desejada e saldo da conta foi actualizado

### Fluxo normal:

1. Cliente indica que pretende levantar dada quantia
2. Máquina MB pergunta se quer talão
3. Cliente responde que não
4. Máquina MB devolve cartão, fornece notas e actualiza saldo
5. Cliente retira cartão e notas

### Fluxo alternativo (1): [cliente quer talão] (passo 3)

- 3.1. Cliente diz que sim
- 3.2. Máquina MB devolve cartão, notas e talão
- 4.3. Cliente retira cartão, notas e talão



## Use Case: Autenticar cliente

**Descrição:** Cliente autentica-se da máquina

**Cenários:** O João levanta €60 com cartão

**Pré-condição:** Nenhum cliente autenticado

**Pós-condição:** Cliente fica autenticado

### Fluxo normal:

1. Cliente apresenta cartão e PIN
2. Máquina MB valida acesso e pede operação

### Fluxo alternativo (1): [cliente autentica-se com MB way] (passo 1)

- 1.1. Cliente escolhe acesso MB way
- 1.2. Máquina MB pede Código MB way
- 1.3. Cliente indica Código MB way
- 1.4. Regressa a 2

### Fluxo de excepção (2): [PIN inválido] (passo 2)

- 2.1. Máquina MB avisa sobre PIN inválido e devolve cartão
- 2.2. Cliente retira cartão



# Diagramas de Use Case - «include»

Opção 1

Use Case: *Levantar €*

...

Fluxo normal:

1. «include» *Autenticar cliente*
3. Cliente indica que pretende levantar dada quantia
3. Máquina MB pergunta se quer talão
4. Cliente responde que não
5. Máquina MB devolve cartão, fornece notas e actualiza saldo
6. Cliente retira cartão e notas

Fluxo alternativo (1): [cliente quer talão] (passo 4)

- 4.1. Cliente diz que sim
- 4.2. Máquina MB devolve cartão, notas e talão
- 4.3. Cliente retira cartão, notas e talão

Use Case: *Pagar serviço*

...

Fluxo normal:

1. «include» *Autenticar cliente*
3. Cliente indica que pretende pagar um serviço

...

Fluxo alternativo (1): ...

...

Use Case: *Efectuar transferência*

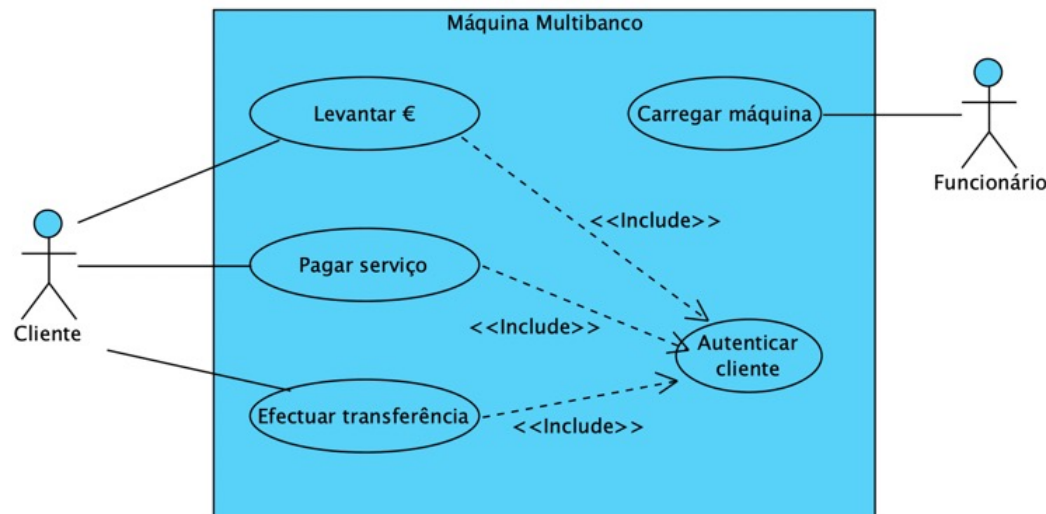
...

Fluxo normal:

1. «include» *Autenticar cliente*
3. Cliente indica que pretende realizar uma transferência...

Fluxo alternativo (1): ...

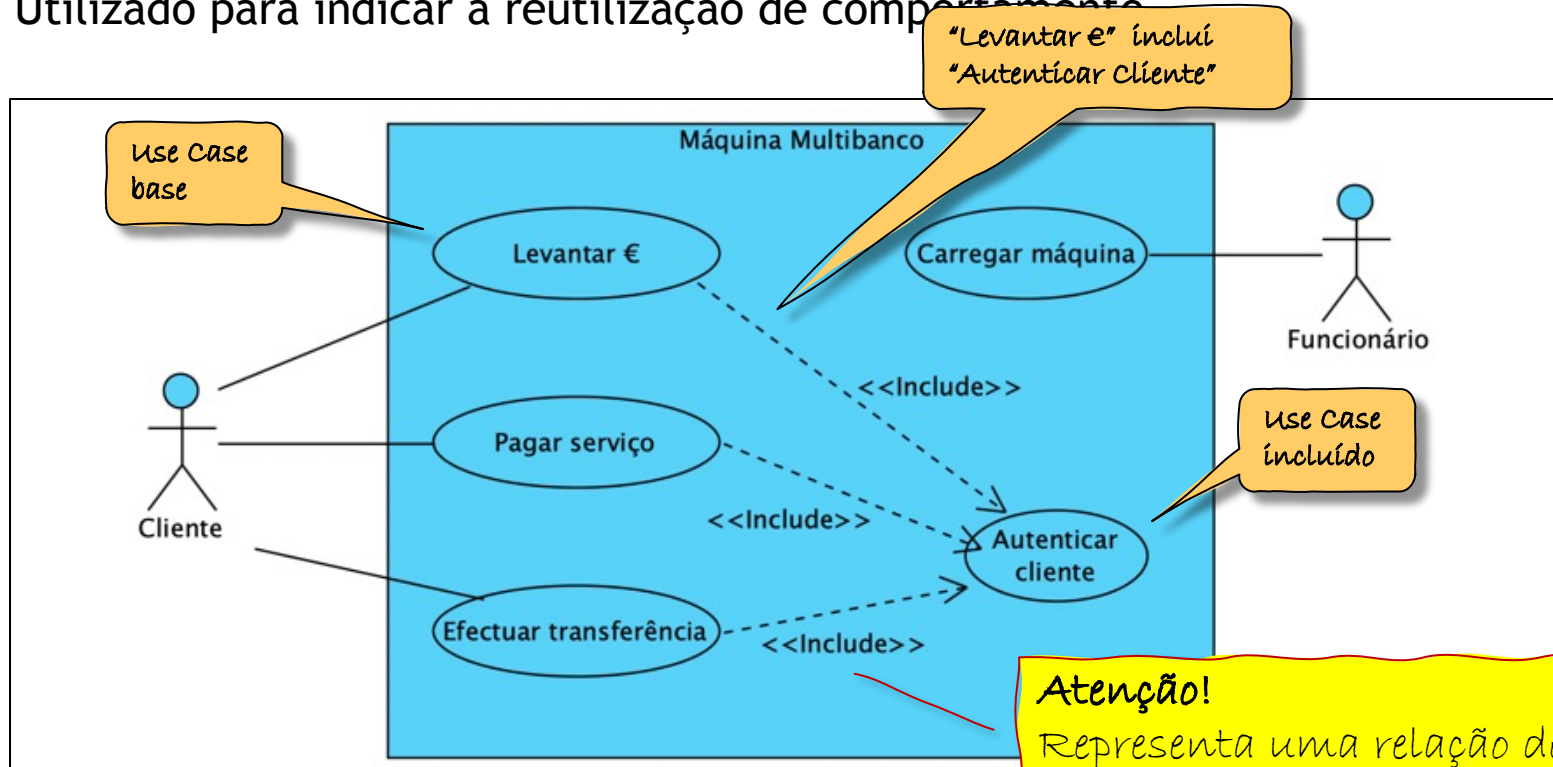
...



# Diagramas de Use Case - «include»

## Opção 1

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



**Atenção!**

Representa uma relação de *inclusão*, não de ordem/sequência!

- Actores utilizam os *Use Case* base.
- Quando o *Use Case* base é executado, também o *Use Case* incluído o é (no ponto definido pelo *Use Case* base)





# Usar pré-condições...

## Opção 2

- Versão alternativa neste caso concreto...
  - Utilizar pré-condições para exigir que o cliente esteja autenticado (com «include» a exigência era que se autenticasse no momento)
  - Semântica do requisito muda!...

Relação não é evidente no diagrama... (mas modelo permite mais flexibilidade no design e implementação do sistema!)

### Use Case: Levantar €

**Descrição:** Cliente levanta quantia da máquina

**Cenários:** O João levanta €60 com cartão; O João levanta €10 com MB way

**Pré-condição:** Sistema tem notas **e Cliente autenticado**

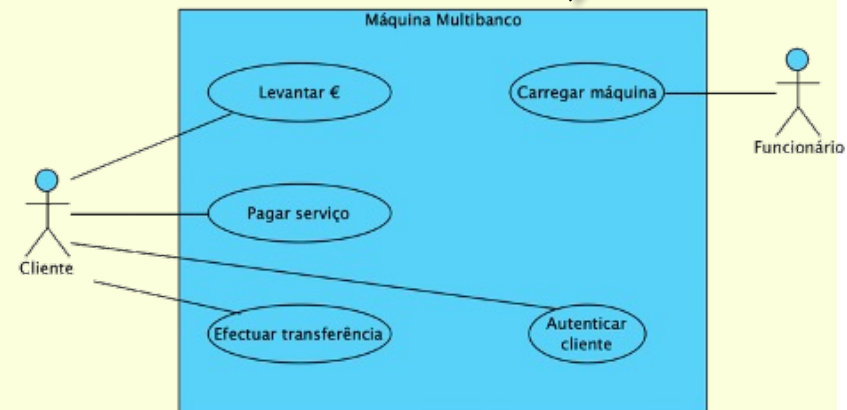
**Pós-condição:** Cliente tem quantia desejada e saldo da conta foi actualizado

#### Fluxo normal:

1. Cliente indica que pretende levantar dada quantia
2. Máquina MB pergunta se quer talão
3. Cliente responde que não
4. Máquina MB devolve cartão e fornece notas
5. Cliente retira cartão e notas

#### Fluxo alternativo (1): [cliente quer talão] (passo 3)

- 3.1. Cliente diz que sim
- 3.2. Máquina MB devolve cartão, notas e talão
- 3.3. Cliente retira cartão, notas e talão





# Diagramas de Use Case - «extend»

## Use Case: Premiar Cliente

**Descrição:** Imprime um talão de prémio para o cliente

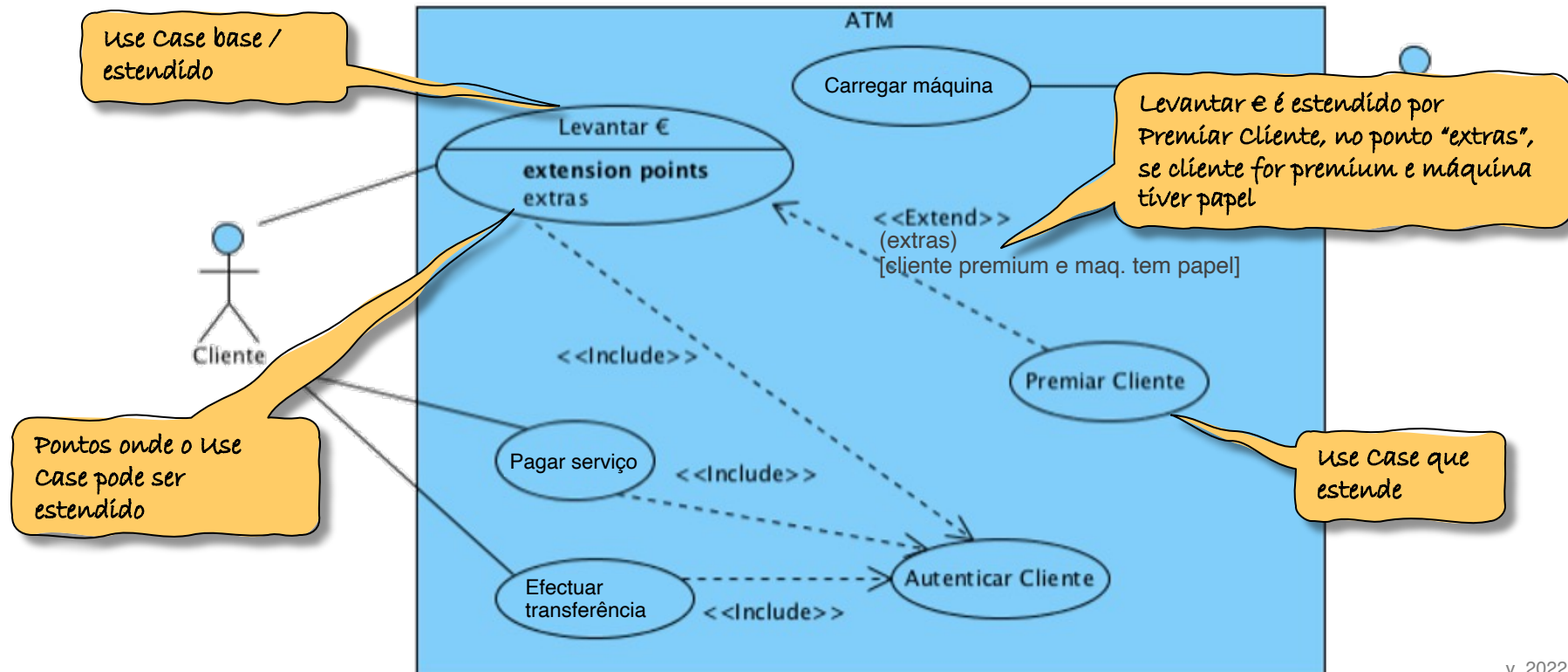
**Cenários:** um novo cenário com emissão de talão de prémio

**Pré-condição:** máquina tem papel

**Pós-condição:** talão é emitido

**Fluxo normal:**

1. Sistema imprime talão com número de prémio
2. Cliente retira o talão







# Diagramas de Use Case - «extend»

## Use Case: Levantar €

**Descrição:** Cliente levanta quantia da máquina

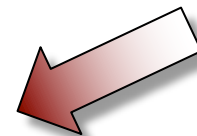
**Cenários:** O João levanta €60 com cartão; O João levanta €10 com MB way

**Pré-condição:** Sistema tem notas

**Pós-condição:** Cliente tem quantia desejada e saldo da conta foi actualizado

### Fluxo normal:

1. «inclui» Autenticar cliente
2. Cliente indica que pretende levantar dada quantia
3. Máquina MB pergunta se quer talão
4. Cliente responde que não
5. Máquina MB devolve cartão e fornece notas *(extras)*
6. Cliente retira cartão e notas



### Fluxo alternativo (1): [cliente quer talão] (passo 4)

- 4.1. Cliente diz que sim
- 4.2. Máquina MB devolve cartão, notas e talão
- 4.3. Cliente retira cartão, notas e talão

- Use Case base não sabe se/por quem é extendido
  - Apenas declara pontos de extensão
  - Extensão é especificada no diagrama

#### Use Case: Premiar Cliente

**Descrição:** Imprime um talão de prémio para o cliente

**Cenários:** um novo cenário com emissão de talão de prémio

**Pré-condição:** máquina tem papel

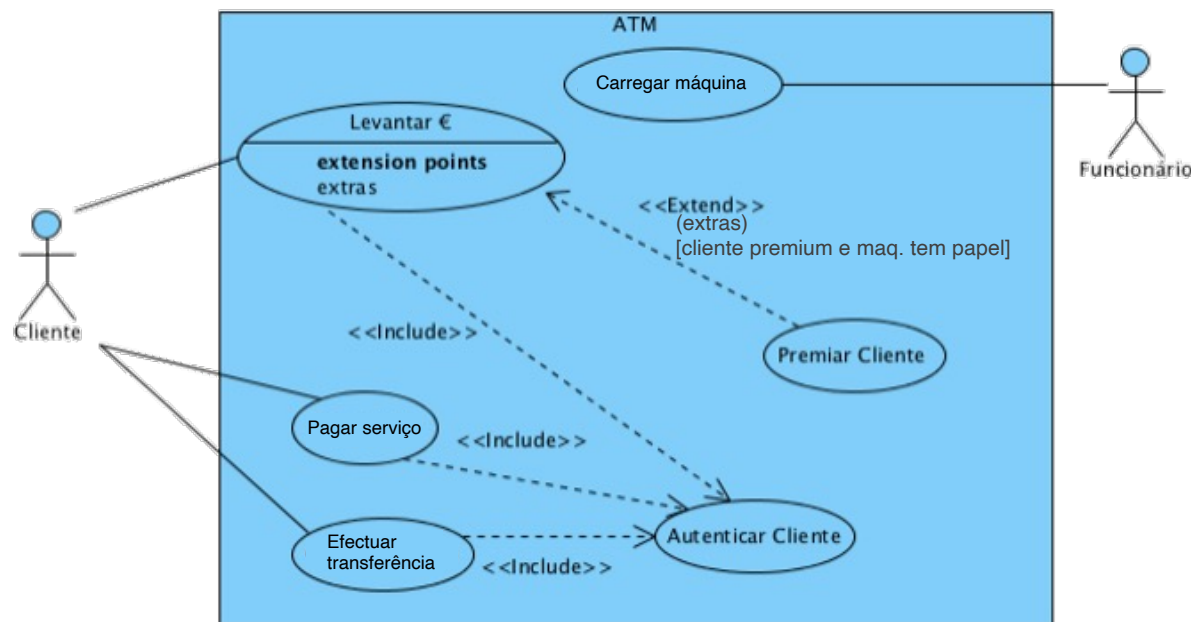
**Pós-condição:** talão é emitido

#### Fluxo normal:

1. Sistema imprime talão com número de prémio
2. Cliente retira o talão

# Diagramas de Use Case - «extend»

- Outro estereótipo de dependência.
- Permite adicionar comportamento a um *use case* base, sob certas condições.



- **Estratégia:**
  - utilizar extensões quando pretendemos reaproveitar/alterar um Use Case base já existente
  - caso base deve ser um *use case* bem formado sem as extensões!
  - extensão pode não ser um *use case* bem formado por si só.



# Generalização/Especialização de Use Cases

## Use Case: Autenticar cliente

**Descrição:** Cliente autentica-se da máquina

**Cenários:** O João levanta €60 com cartão

**Pré-condição:** Nenhum cliente autenticado

**Pós-condição:** Cliente fica autenticado

**Fluxo normal:**

1. Cliente apresenta cartão e PIN
2. Máquina MB valida acesso e pede operação

**Fluxo alternativo (1):** [cliente autentica-se com MB way] (passo 1)

- 1.1. Cliente escolhe acesso MB way
- 1.2. Máquina MB pede Código MB way
- 1.3. Cliente indica Código MB way
- 1.4. Máquina MB valida acesso e pede operação

**Fluxo de exceção (1):**

[falhas de acesso]

- 2.1. Cliente fornece cartão
- 2.2. Máquina MB valida acesso e pede operação

**Fluxo de exceção (2):**

[falhas de acesso]

- 2.1. Cliente fornece cartão

## Use Case: Autenticar com Cartão

**Descrição:** Cliente autentica-se da máquina com cartão

**Cenários:** Todos

**Pré-condição:** Nenhum cliente autenticado

**Pós-condição:** Cliente fica autenticado

**Fluxo normal:**

1. Cliente apresenta cartão e PIN
2. Máquina MB valida acesso e pede operação

**Fluxo de exceção (1):** [PIN inválido e nº de tentativas falhadas seguidas < 3] (passo 2)

- 2.1. Máquina MB avisa sobre PIN inválido e fornece cartão
- 2.2. Cliente retira cartão

**Fluxo de exceção (2):** [PIN inválido e nº de tentativas falhadas seguidas = 3] (passo 2)

- 2.1. Máquina MB avisa sobre PIN inválido retém cartão

Use Case **não** está completo.

P.e.: Código MBway errado?!

Autenticações com PIN e com MBway correspondem a cenários distintos! Complicado colocar no mesmo Use Case...

## Use Case: Autenticar com telemóvel

**Descrição:** Cliente autentica-se da máquina com MB way

**Cenários:** O João levanta €60 com cartão

**Pré-condição:** Nenhum cliente autenticado

**Pós-condição:** Cliente fica autenticado

**Fluxo normal:**

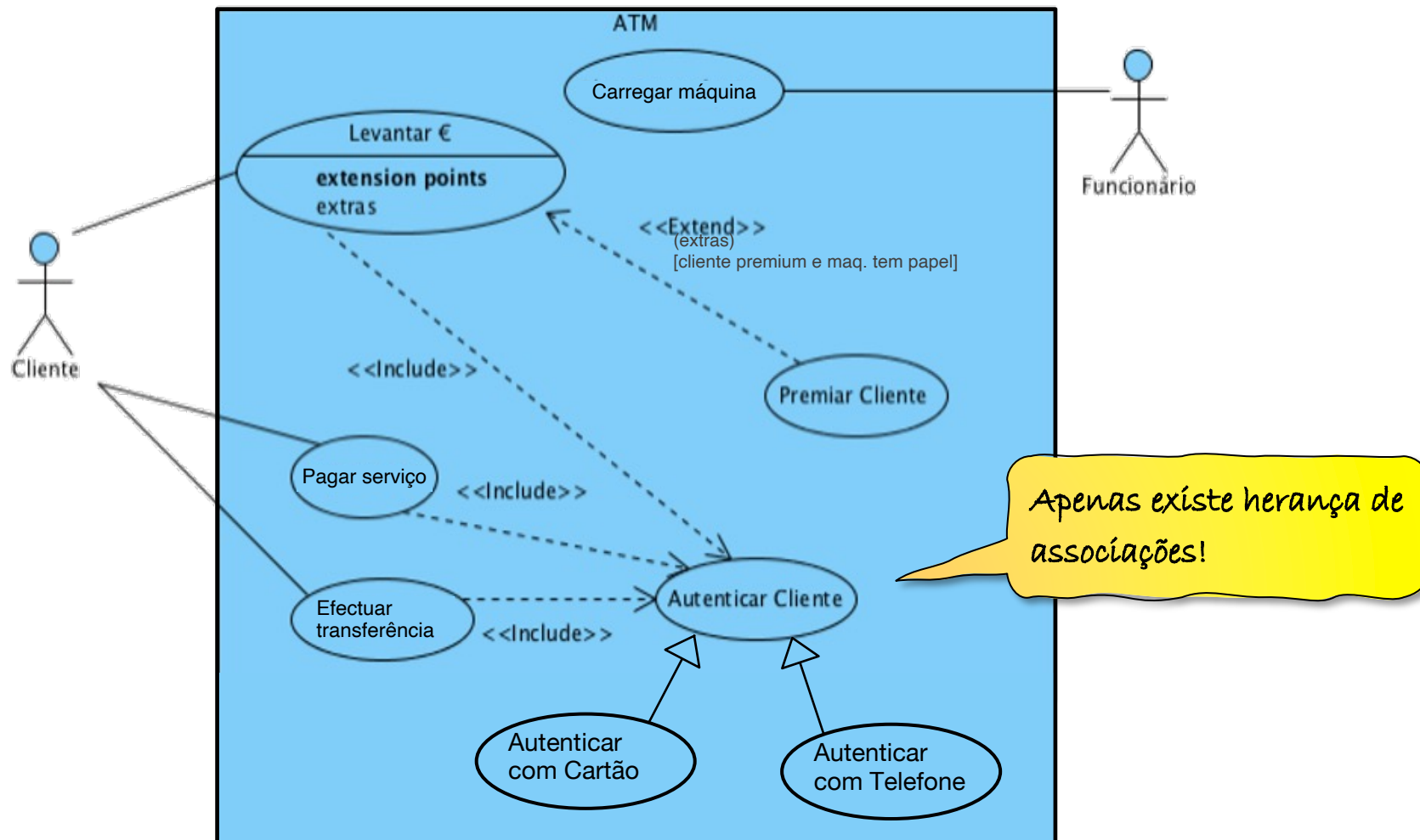
1. Cliente escolhe acesso MB way
2. Máquina MB pede Código MB way
3. Cliente indica Código MB way
4. Máquina valida acesso e pede operação

**Fluxo de exceção (1):** [Código MB way inválido] (passo 3)

- 3.1. Máquina MB avisa sobre Código inválido



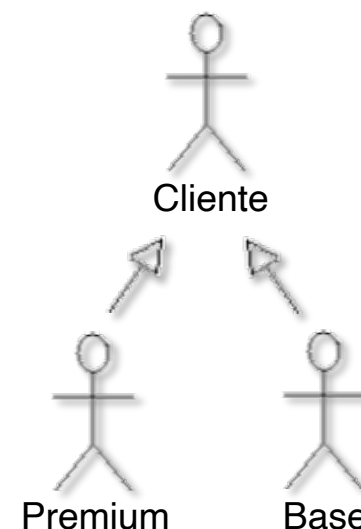
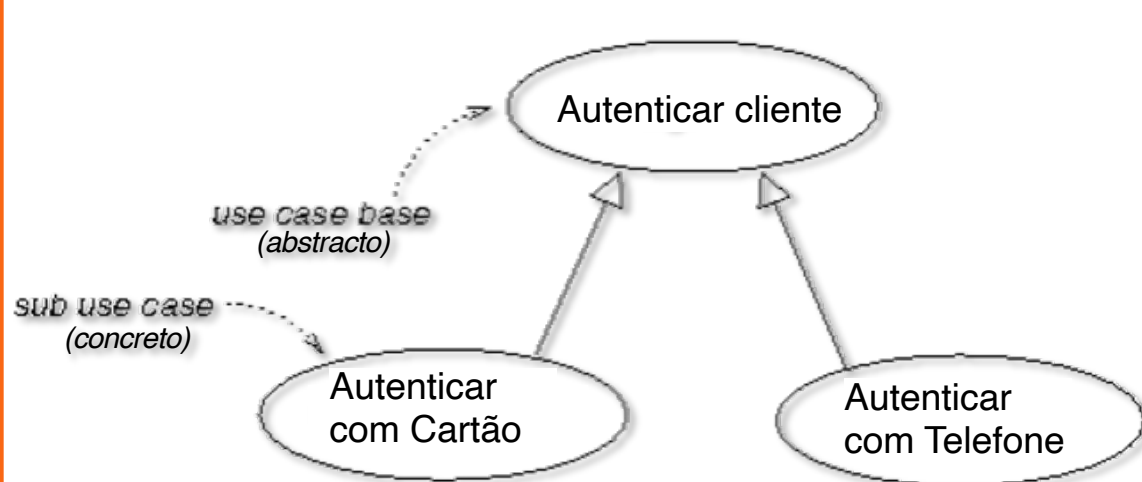
# Generalização/Especialização de Use Cases



- O cliente pode realizar qualquer uma das formas de autenticação.



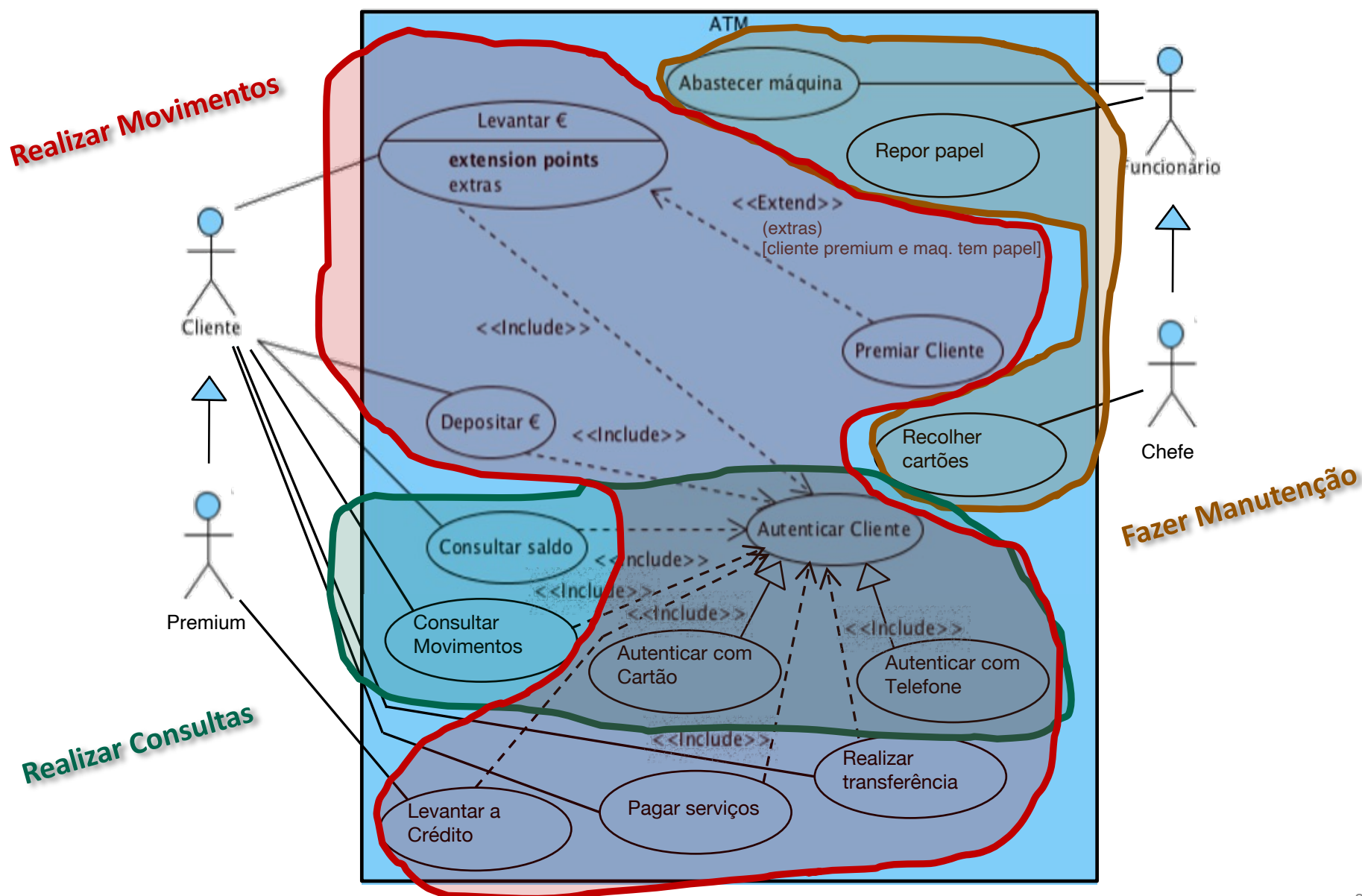
# Diagramas de UC - Generalização/Especialização



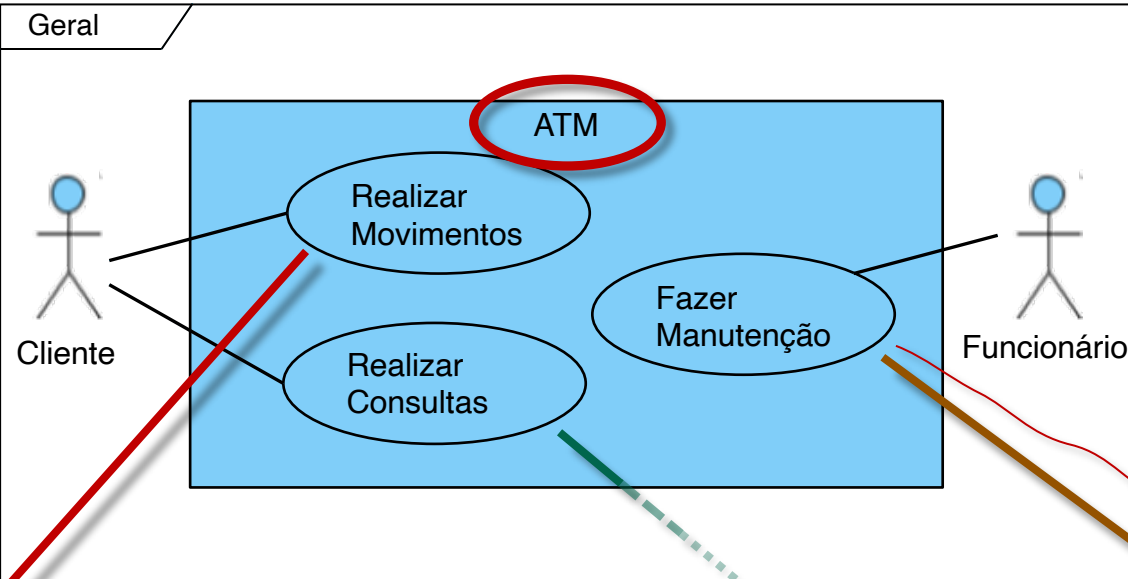
- Sub-elementos são casos particulares de super-elementos.
- Um sub-elemento pode ser utilizado onde quer que o super-elemento possa.
- No caso dos Use Case, útil para organizar Use Cases por tipo.
- No caso dos Actores, útil para user profiling (definição de níveis de acesso).
- Nos exemplos apresentados:
  - Existem duas formas de um cliente se autenticar.
  - Os actores Premium e Base podem realizar todos os use cases de Cliente (e cada um pode ter Use case específicos).



# Estruturação de modelos?





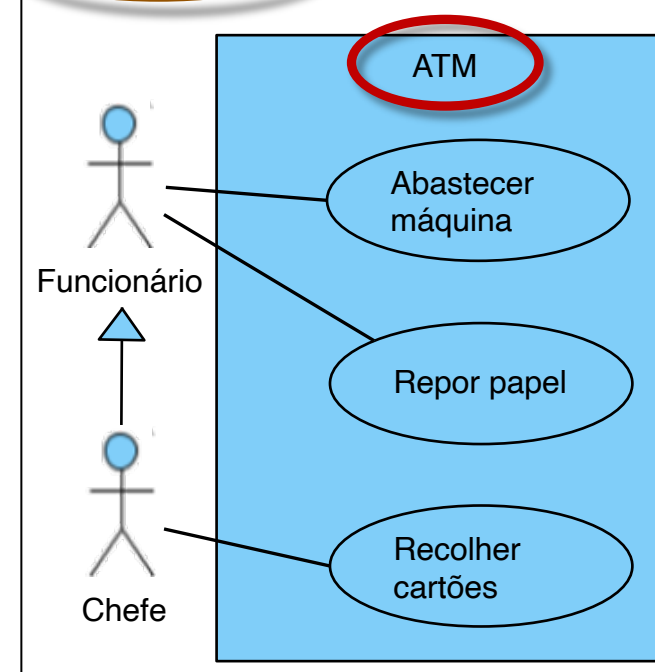
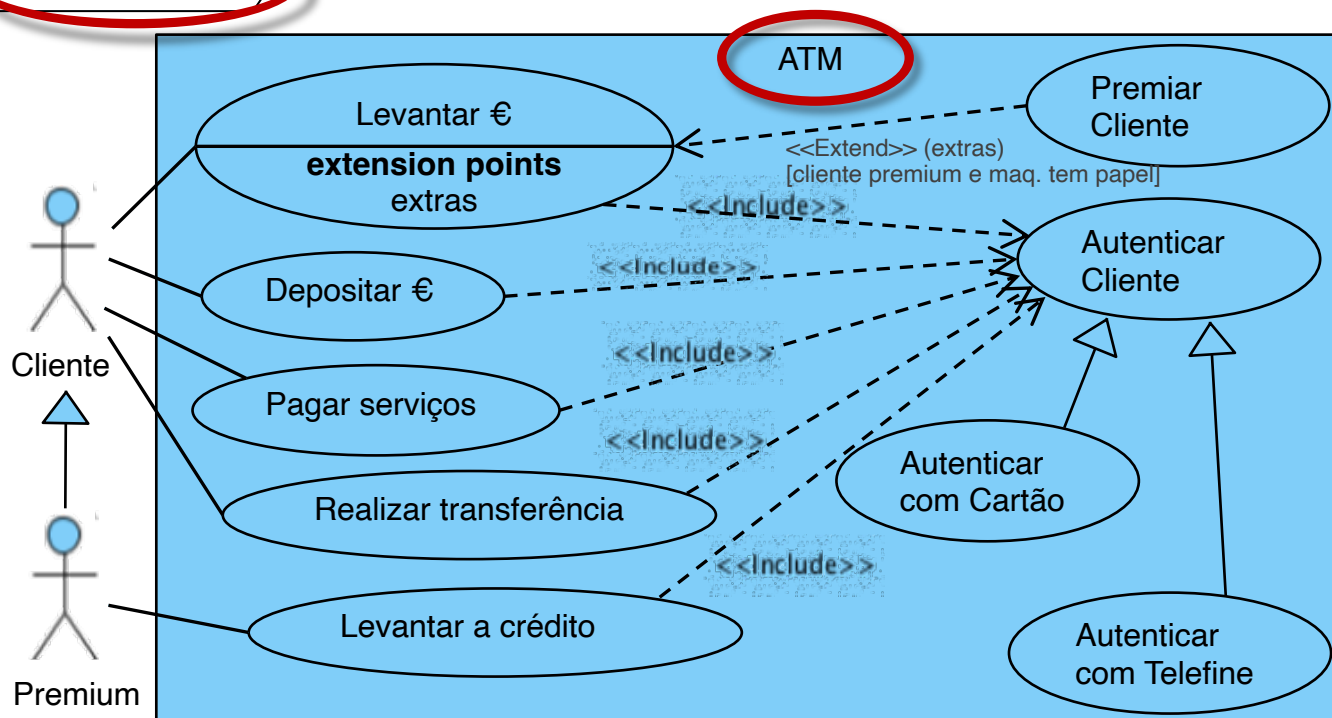


Vários diagramas,  
um modelo!

Não são especificados!

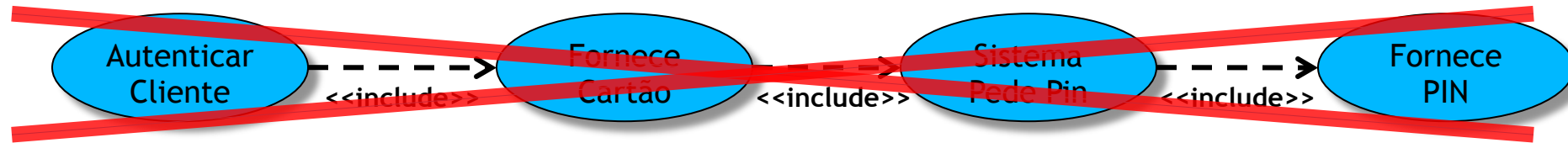
Realizar Movimentos

Fazer Manutenção

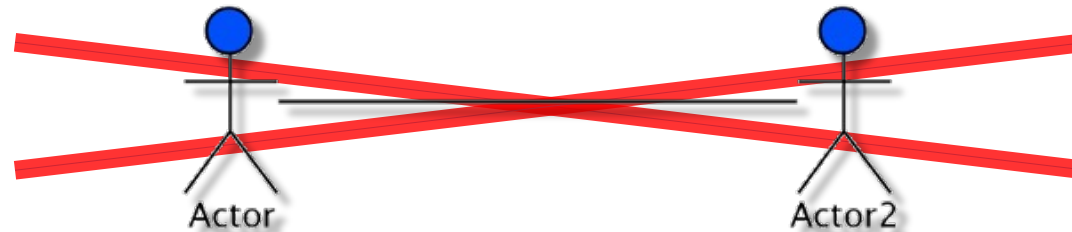


# Alguns aspectos a ter em atenção...

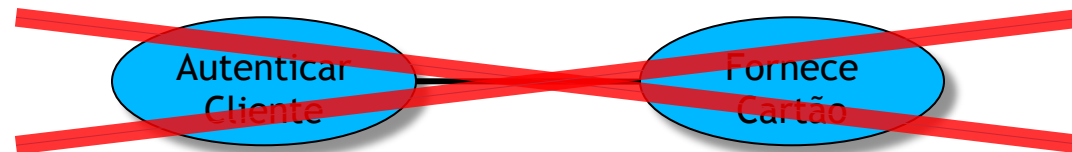
- Modelo de Use Cases não representa fluxo de dados/controlo



- Modelo de Use Cases não representa comunicação entre actores



- Modelo de Use Cases não representa comunicação entre use cases



- «include», «extend» e generalização devem ser utilizados com muita moderação!





# Alguns aspectos a ter em atenção...

- Estruturas de controlo: if ... then ... else ...

1. «include» Autenticar cliente
2. Cliente indica que pretende levantar dada quantia
3. Máquina MB pergunta se quer talão
4. **se** Cliente responde que não  
**então** Sistema devolve cartão, fornece notas e actualiza saldo  
**senão** Sistema devolve cartão, notas e talão
5. Cliente retira ???

- Vantagens

- Estrutura de controlo familiar
- Diminui número de fluxos alternativos

- Inconvenientes

- Use Cases mais difíceis de ler/compreender
- Cenários alternativos mais difíceis de identificar
- Use Cases mais difíceis de testar e de implementar

- Estruturas de controlo: go to... (regressar a...)

- permitem definir ciclos - evitar!

Use Case: Levantar €

...

Fluxo normal:

1. «include» Autenticar cliente
3. Cliente indica que pretende levantar dada quantia
3. Sistema pergunta se quer talão
4. Cliente responde que não
5. Sistema devolve cartão, fornece notas e actualiza saldo
6. Cliente retira cartão e notas

Fluxo alternativo (1): [cliente quer talão] (passo 4)

- 4.1. Cliente diz que sim
- 4.2. Sistema devolve cartão, notas e talão
- 4.3. Cliente retira cartão, notas e talão



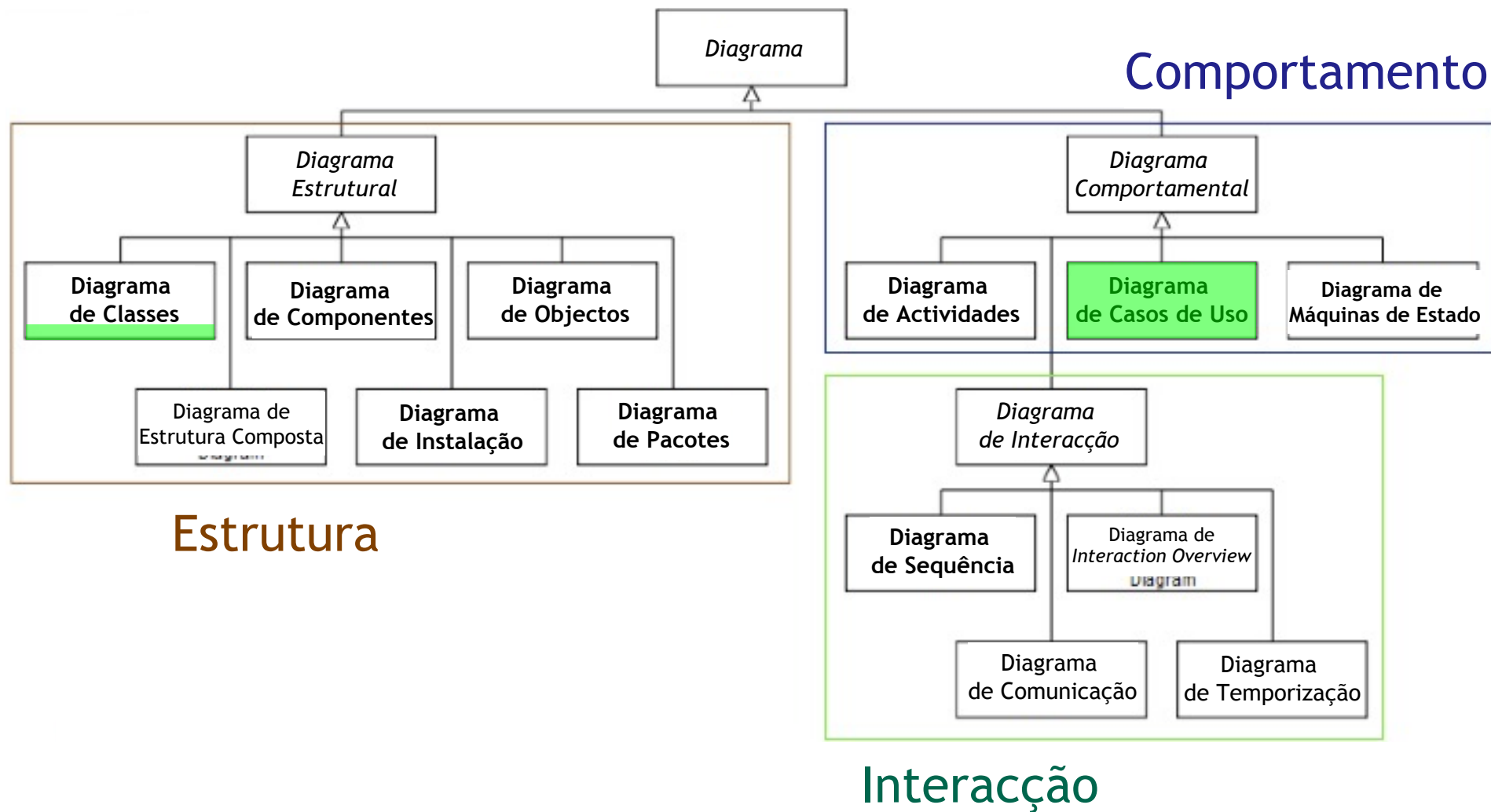


# Resumindo

- Um Use Case descreve as **sequências de interações** entre actores externos e um sistema em projecto (visto como uma *black box*), para que um dado cenário de utilização seja suportado.
- Cada passo de interação actor-sistema descrito num Use Case designa-se por *evento*, *acção* ou *operação*, e deve identificar a sua origem (actor ou sistema);
- Um Use Case descreve um fluxo principal de eventos/operações, designado *fluxo principal*, bem como outros possíveis fluxos ou caminhos designados *fluxos alternativos*, bem ainda como fluxos que conduzem a situações de insucesso (*fluxos de excepção*);
- Use Cases podem ser especificados recorrendo a outros Use Cases, através de **relações de inclusão, extensão e generalização**;
- A **generalização** é também aplicável aos actores, desta forma sendo possível representar o relacionamento entre actores/papéis perante o sistema;
- Use Cases devem ser simples e legíveis, **não devem conter detalhes sobre a interface** com o utilizador e devem ter o nível de detalhe necessário a cada iteração de requisitos (são refináveis);
- Use Cases relacionados com actores devem ser identificados por **verbos no infinitivo**, deixando claro qual a tarefa que o sistema deve fornecer ao actor.

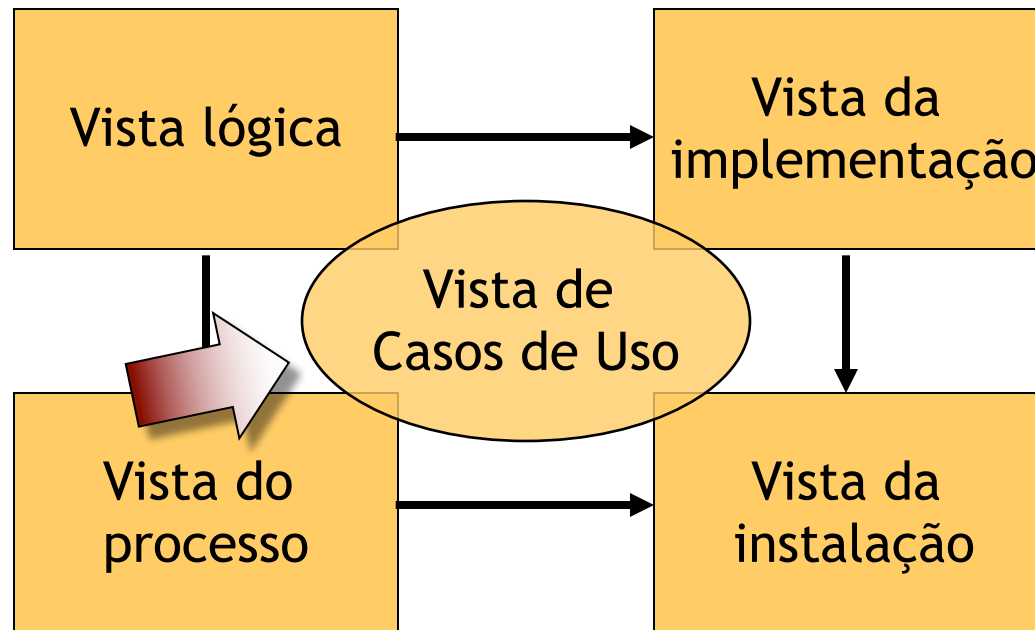


# Diagramas da UML 2.x





# Onde estamos...



(Kruchten, 1995)



# *Use Cases*

*“To my knowledge, no other software engineering language construct as significant as use cases has been adopted so quickly and so widely among practitioners. I believe this is because use cases play a role in so many different aspects of software engineering”*

*Use Cases - Yesterday, Today, and Tomorrow*

Ivar Jacobson (*‘pai’ dos Use Cases*)

(e também, Vice President  
Process Strategy

Rational Software  
IBM Software Group)