

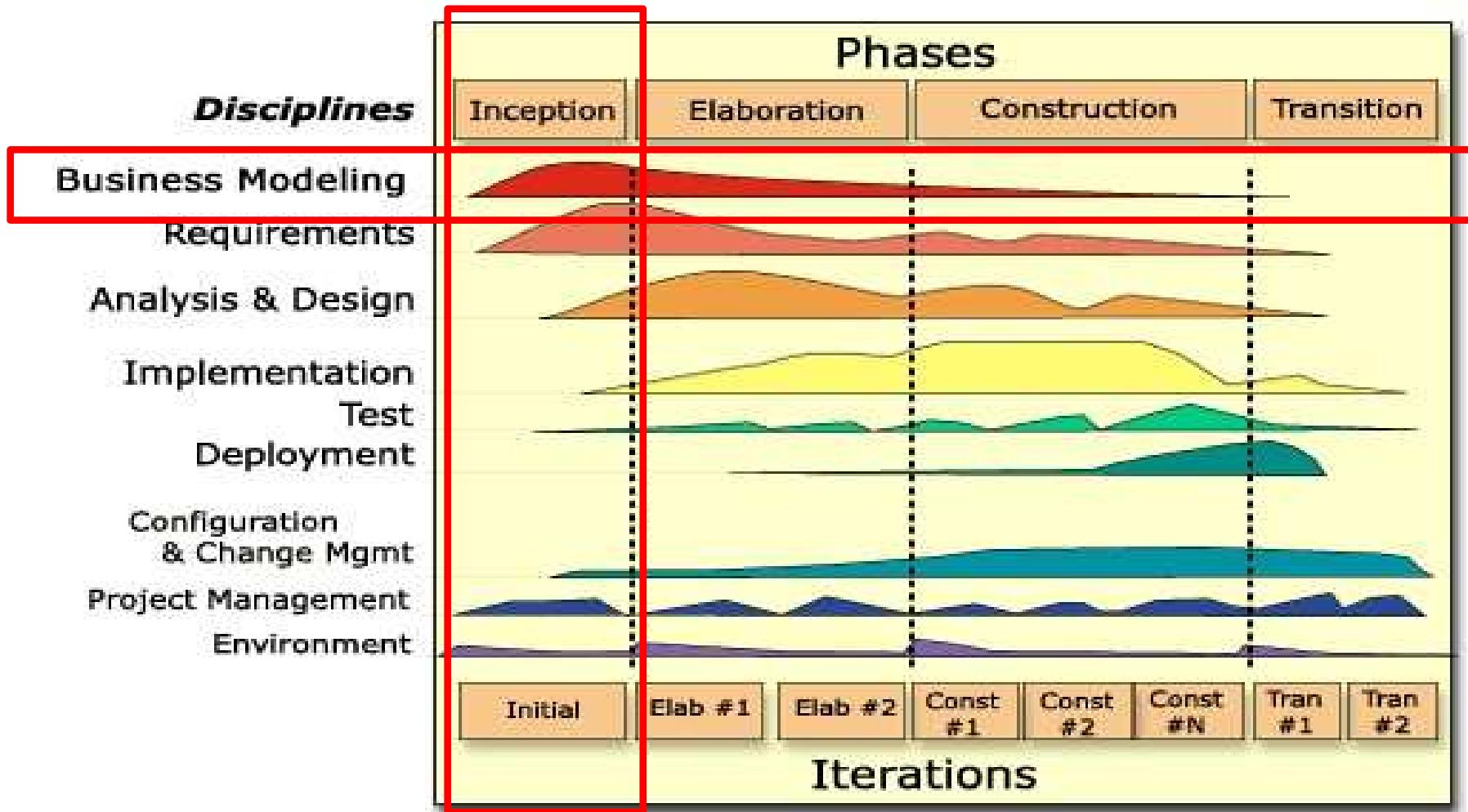


# Desenvolvimento de Sistemas Software

Aula Teórica 4

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

# Próximos passos...



# Tipos de requisitos

- Requisitos funcionais - O que o sistema deve fazer
  - Descrevem as interacções entre o Sistema e o seu ambiente
    - Descrições independentes da implementação
    - Ambiente: Utilizadores e outros sistemas (externos)
- Requisitos não funcionais - Como o sistema deve fazê-lo
  - Aspectos do sistema não directamente relacionados com o seu comportamento funcional
    - *Usability; Dependability*<sub>(reliability, robustness, safety)</sub>; *Performance; Maintainability; Portability*; e ainda: de implementação, de interface (com outros sistemas), de operação, de modo de entrega, legais.



# Validação dos requisitos

- Os requisitos são **validados com o cliente**
  - Validação deve considerar:
    - **Completude** - todos os aspectos relevantes foram considerados
    - **Consistência** - não existem contradições entre requisitos
    - (ausência de) **Ambiguidade** - nenhum requiseito pode ser interpretado de formas diferentes
    - **Correcção** - os requisites descrevem correctamente o que o cliente pretende e o que a equipa de desenvolvimento se propõe fazer
- A especificação de requisitos deve ainda ser...
- Realista - não se deve prometer o que não podemos realizar
  - Verificável - temos que poder saber se cumprimos os objectivos!
  - Rastreável - porque é que cada requisito lá está?

# Definição de requisitos (funcionais)

Definição de requisitos do sistema, duas abordagens possíveis:

- Foco no Sistema (Visão estrutural - interna)
- Foco no Utilizador (Visão orientada aos use case - externa)

## Foco no Sistema (POO)

1. Definir classes;
2. Definir métodos das classes;
3. Definir interface com o utilizador (comportamento do sistema face ao utilizador);

*Problemas:* O que interessa ao utilizador é o comportamento do sistema, no entanto a interface com o utilizador só é definida no final do processo.

- Perigo de o sistema não fornecer toda a funcionalidade pretendida;
- Perigo de o sistema fornecer funcionalidade não pretendida  
(= despedício de trabalho).

# Definição de requisitos (funcionais)

## Visão orientada aos *Use Case*

1. Identificar **Cenários** - descrição informal, mas concreta e focada, de uma interacção com o Sistema do ponto de vista de um utilizador
  - Ajudam a análise pois são comprehensíveis para os clientes
2. Identificar e especificar os ***Use Cases*** (Casos de Uso) - descrições de fluxos de interacção com o Sistema para suportar os cenários
  - Cenários são instâncias dos *Use Case*
3. Identificar **Actores** - entidades externas que interagem com o sistema
4. Identificar relações entre Actores e *Use Cases*
5. Identificar componentes de suporte à realização dos *Use Case* - como vai a funcionalidade necessária ser implementada

### Vantagens:

- Não há trabalho desnecessário.
- O Sistema de Informação suporta as tarefas do cliente.
- As fronteiras do Sistema ficam bem definidas.

# Definição de Use Case

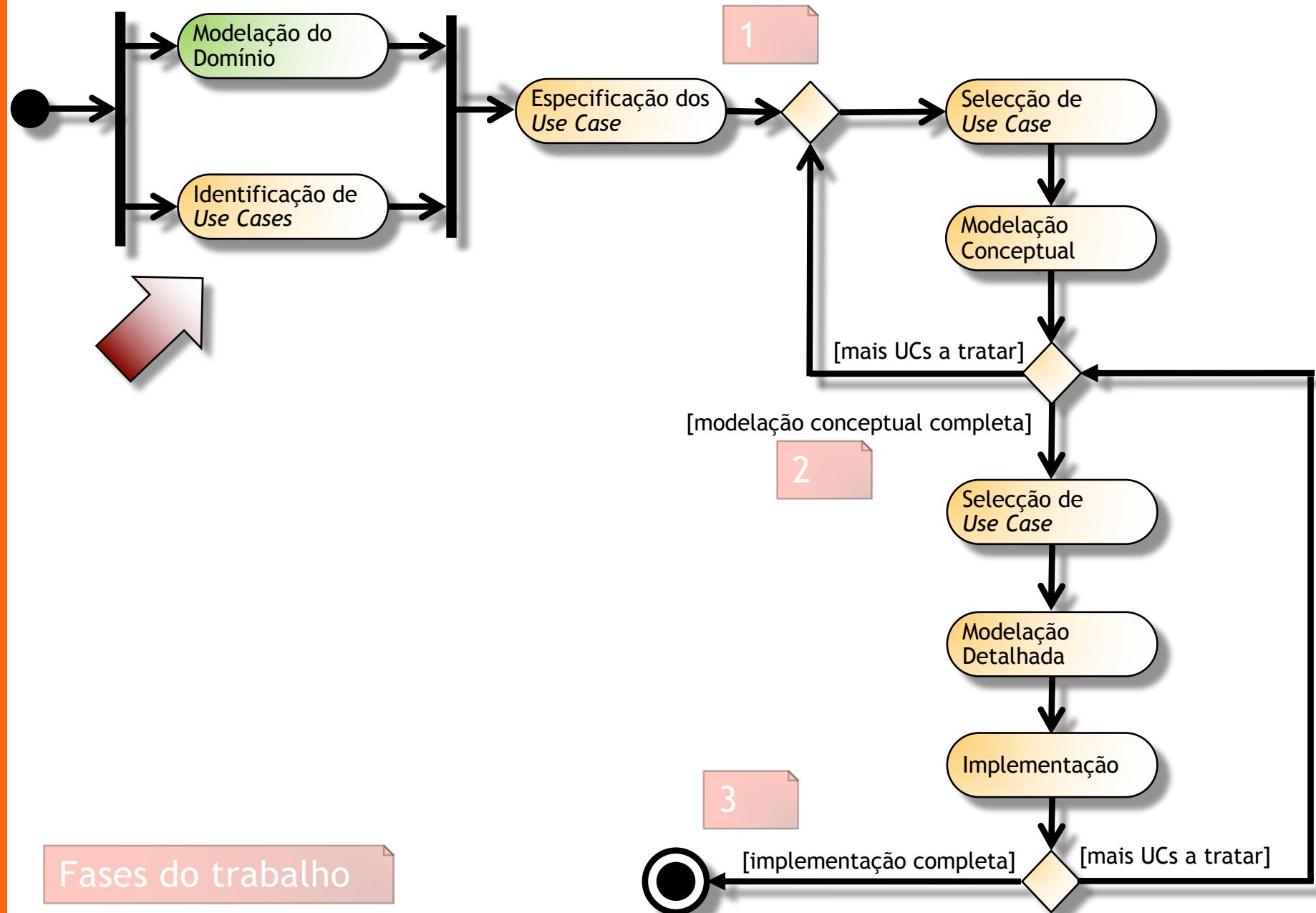
- Descreve como os Actores atingem objectivos (realizam os *Use Cases*) utilizando o sistema
  - Definem relação entre *inputs* dos Actores e comportamento do Sistema
- Especificação deve incluir o comportamento tipicamente esperado, bem como variantes:
  - Comportamentos alternativos que ainda levam ao sucesso
  - Comportamentos de insucesso (Excepções)
  - vamos também definir as pré-condições e pós-condições de cada use case (cf. *design by contract*).



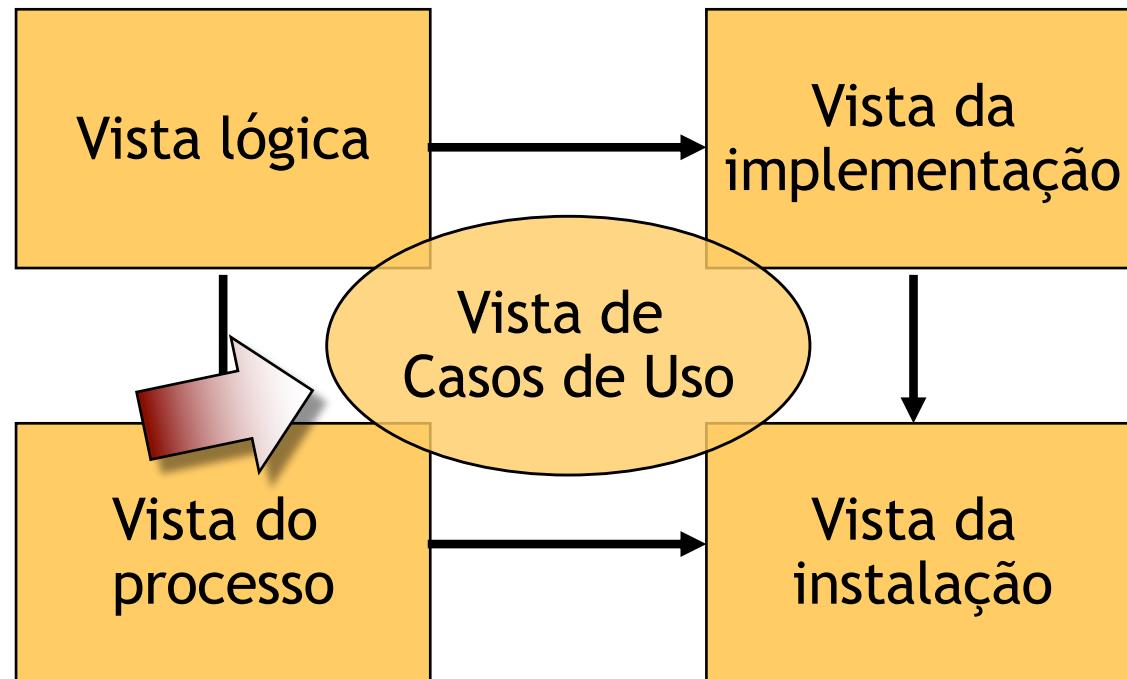
## *Design by contract*

- *Design by contract* (DBC) baseia-se na noção de um contrato entre um cliente e um fornecedor para a realização de um serviço.
- O conceito central do DBC é a asserção (uma asserção é uma expressão booleana que nunca deverá ser falsa).
- Tipicamente as asserções são automaticamente testadas durante a fase de *debug*.
- O DBC identifica três tipos de asserções:
  - pré-condições - condições que se devem verificar para a invocação de um dado serviço ser válida;
  - pós-condições - condições que se devem verificar após a execução de um serviço;
  - invariantes - asserções que se devem verificar durante o tempo de vida da entidade a que se aplicam.
- A partir da versão 1.4 o Java passou a ter *asserts* que podem ser utilizados para definir pré- e pós-condições - no entanto não suporta invariantes.

# Onde estamos...



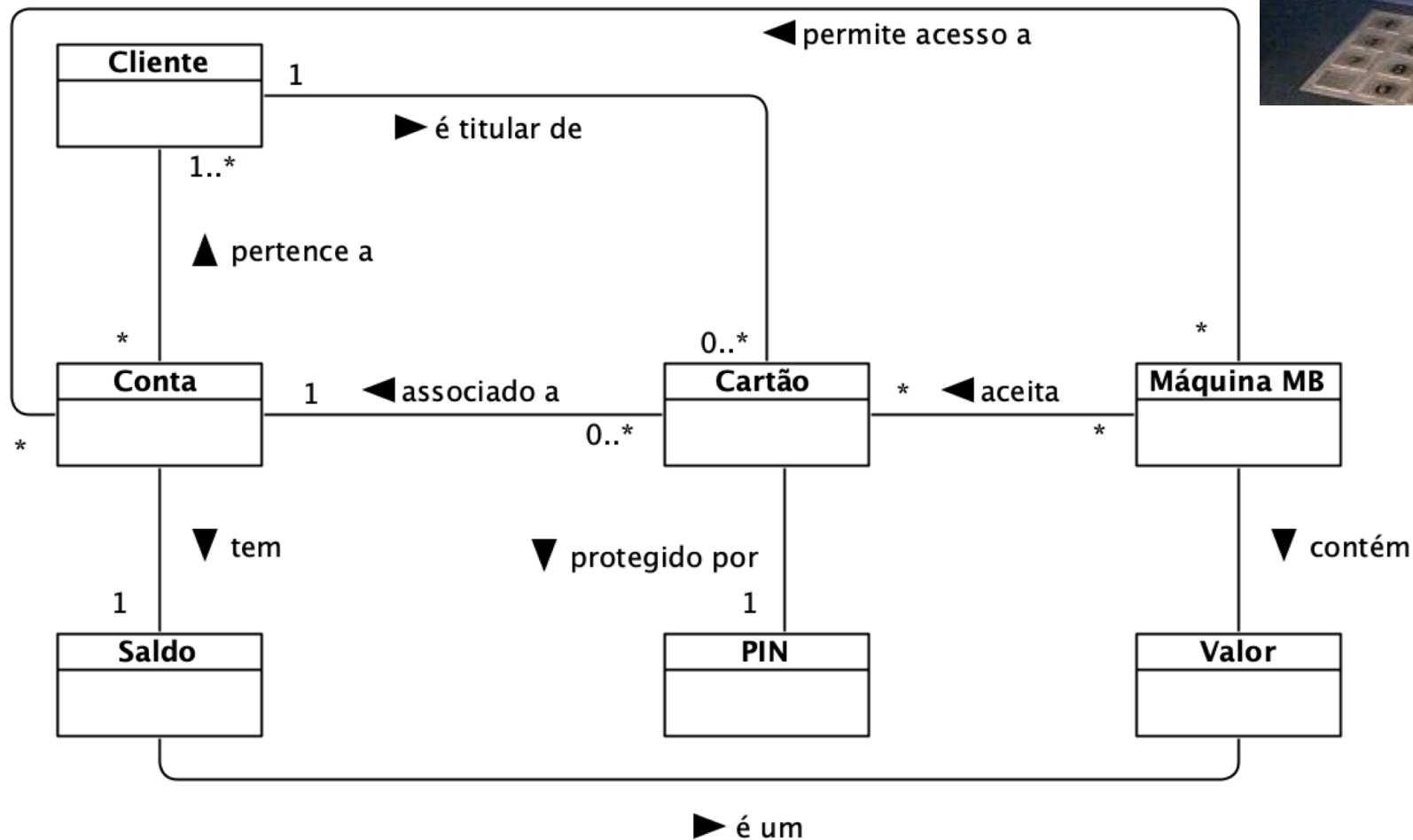
# Onde estamos...



(Kruchten, 1995)

# Exemplo - Máquina Multibanco

- Modelo de Domínio



# Exemplo - Máquina Multibanco

- **Cenários**

1. O João levanta €60 com cartão

O João vai viajar e dirige-se a uma máquina MB para levantar dinheiro para a viagem. Introduz o cartão e o código PIN quando tal é solicitado pela máquina. No menu escolhe a opção de levantamento de €60. A máquina pergunta ao João se pretende um talão e ele responde que não. A máquina disponibiliza então o valor pedido e o João retira-o. Finalmente, a máquina devolve o cartão, que o João recolhe também.

2. O João levanta €10 com MB way

...

3. A Maria paga a conta da luz

...

4. O Rui transfere a mesada para a conta da filha

...

5. A Joana abastece a máquina com notas

...

6. ...



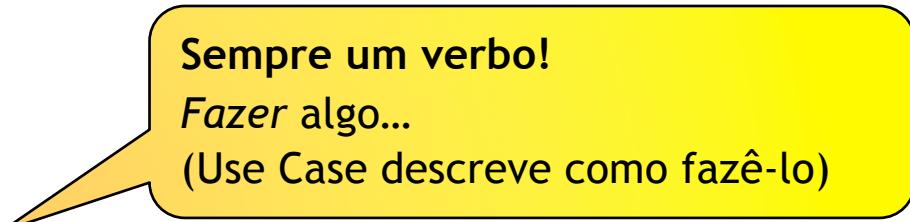
# Exemplo - Máquina Multibanco

- **Cenários**

1. O João levanta €60 com cartão
2. O João levanta €10 com MB way
3. A Maria paga a conta da luz
4. O Rui transfere a mesada para a conta da filha
5. A Joana abastece a máquina com notas
6. ...

- **Use Cases**

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



Sempre um verbo!  
*Fazer* algo...  
(Use Case descreve como fazê-lo)

- **Actores**

- Cliente (cf. João, Maria, Rui)
- Bancário (cf. Joana)

# Definição de Use Case - Tipos de fluxos

- Em cada especificação de um *Use Case* podem/devem existir diferentes fluxos de controlo (sequências de eventos, comportamentos)
- Podemos caracterizá-los em três tipos:
  - **Fluxo Normal** (ou Principal)  
O fluxo mais comum. Representa uma situação perfeita em que nada corre mal.  
A pós-condição é satisfeita no final (se pré-condição também o é no início).
  - **Fluxos Alternativos**  
Fluxos válidos mas menos comuns.  
A pós-condição é satisfeita (se pré-condição também o é no início)
  - **Fluxos de Excepção**  
Condições de erro suficientemente importantes para serem capturadas no modelo.  
A pós-condição NÃO é satisfeita.

# Definição de Use Case - Especificação

- A UML não especifica como descrever *Use Cases*
  - *Tem que ser definido por cada organização ou projecto*
  - *Muitas abordagens são possíveis/comuns*
    - *Desde descrições textuais até especificações via diagramas*
    - *Mais ou menos verbosas e detalhadas*
- Em DSS vamos utilizar uma notação semi-estruturada:
  - **Use Case:** nome do use case
    - **Descrição:** breve descrição do use case
    - **Cenários:** cenários que originam o use case
    - **Pré-condição:** o que deve ser verdade no Sistema para que executar o use case seja válido
    - **Pós-condição:** condição de sucesso do use case
    - **Fluxo normal:**
      - Fluxo de eventos mais comum
    - **Fluxos alternativos:**
      - Especificação dos modos alternativos de completar o use case
    - **Fluxos de exceção:**
      - Especificação de situações relevantes em que o use case não tem sucesso

(cf. Verificável!)

# Exemplo - Máquina Multibanco

- **Use Case:** Levantar €
  - **Descrição:** Cliente levanta quantia da máquina
  - **Cenários:** O João levanta €60 com cartão
  - **Pré-condição:** Sistema tem notas
  - **Pós-condição:** Cliente tem quantia desejada e saldo da conta foi actualizado
  - **Fluxo normal:**
    - Cliente levanta dinheiro com cartão e PIN, sem pedir talão
  - **Fluxos alternativos:**
    - Cliente pede talão
    - Cliente autentica-se com MB way
    - ...?
  - **Fluxos de excepção:**
    - Autenticação inválida
    - Máquina MB não tem valor suficiente
    - ...?

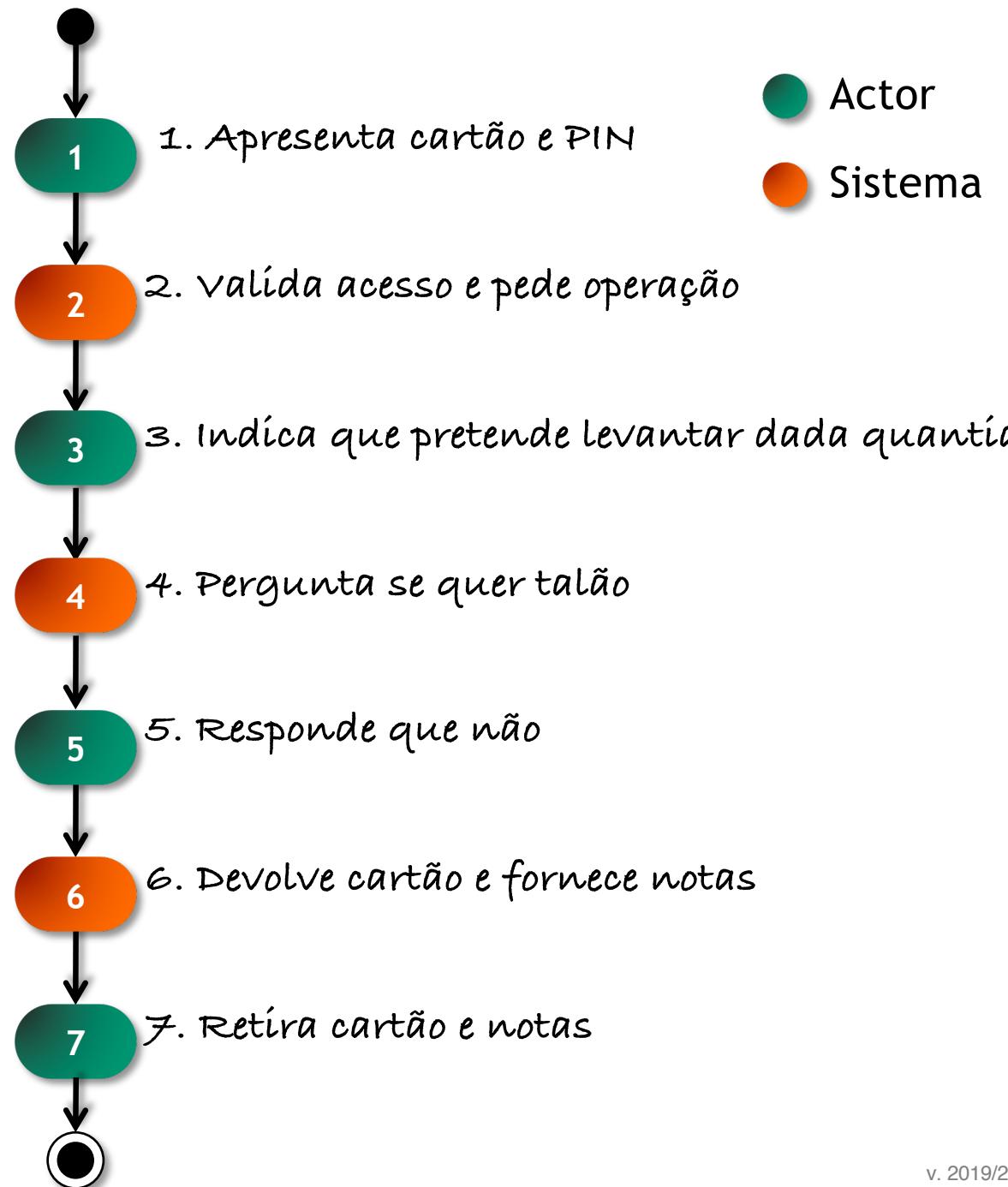
# Exemplo - Máquina Multibanco

- **Use Case:** Levantar €
  - **Descrição:** Cliente levanta quantia da máquina
  - **Cenários:** O João levanta €60 com cartão
  - **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 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 e fornece notas
    7. Cliente retira cartão e notas
  - **Fluxo alternativo 1:** [Cliente quer talão] (passo 5)
    - 5.1. Cliente diz que sim
    - 5.2. Máquina MB devolve cartão, notas e talão
    - 5.3. Cliente retira cartão, notas e talão
  - **Fluxo de excepção 1:** [PIN inválido] (passo 6)
    - 6.1. Máquina MB avisa sobre PIN inválido e fornece cartão
    - 6.2. Cliente retira cartão

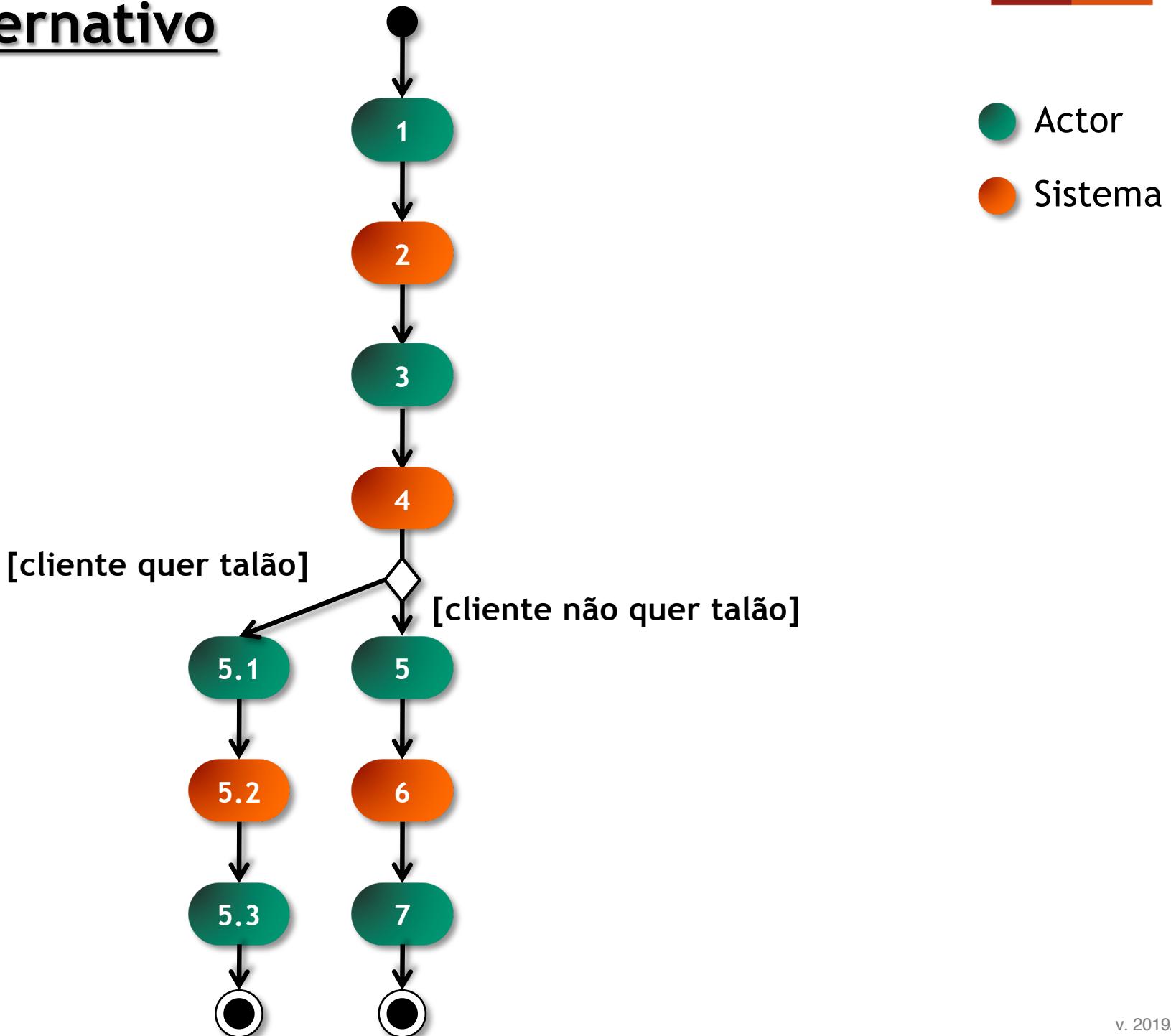
# Exemplo - Máquina Multibanco

- **Use Case:** Levantar €
  - **Descrição:** Cliente levanta quantia da máquina
  - **Cenários:** O João levanta €60 com cartão
  - **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 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 e fornece notas~~
    7. ~~Cliente retira cartão e notas~~
  - **Fluxo alternativo 1:** [Cliente quer talão] (passo 5)
    - 5.1. Cliente diz que sim
    - 5.2. Máquina MB devolve cartão, notas e talão
    - 5.3. Cliente retira cartão, notas e talão
  - **Fluxo de excepção 1:** [PIN inválido] (passo 6)
    - 6.1. Máquina MB avisa sobre PIN inválido e fornece cartão
    - 6.2. Cliente retira cartão

## Fluxo normal



## Fluxo alternativo



# Exemplo - Máquina Multibanco

- **Use Case:** Levantar €

- **Descrição:** Cliente levanta quantia da máquina
- **Cenários:** O João levanta €60 com cartão
- **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 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 e fornece notas~~
7. ~~Cliente retira cartão e notas~~

Não termina com sucesso.  
Pós-condição não se verifica

- **Fluxo alternativo 1:** [Cliente quer talão] (passo 5)

- 5.1. Cliente diz que sim
- 5.2. Máquina MB devolve cartão, notas e talão
- 5.3. Cliente retira cartão, notas e talão

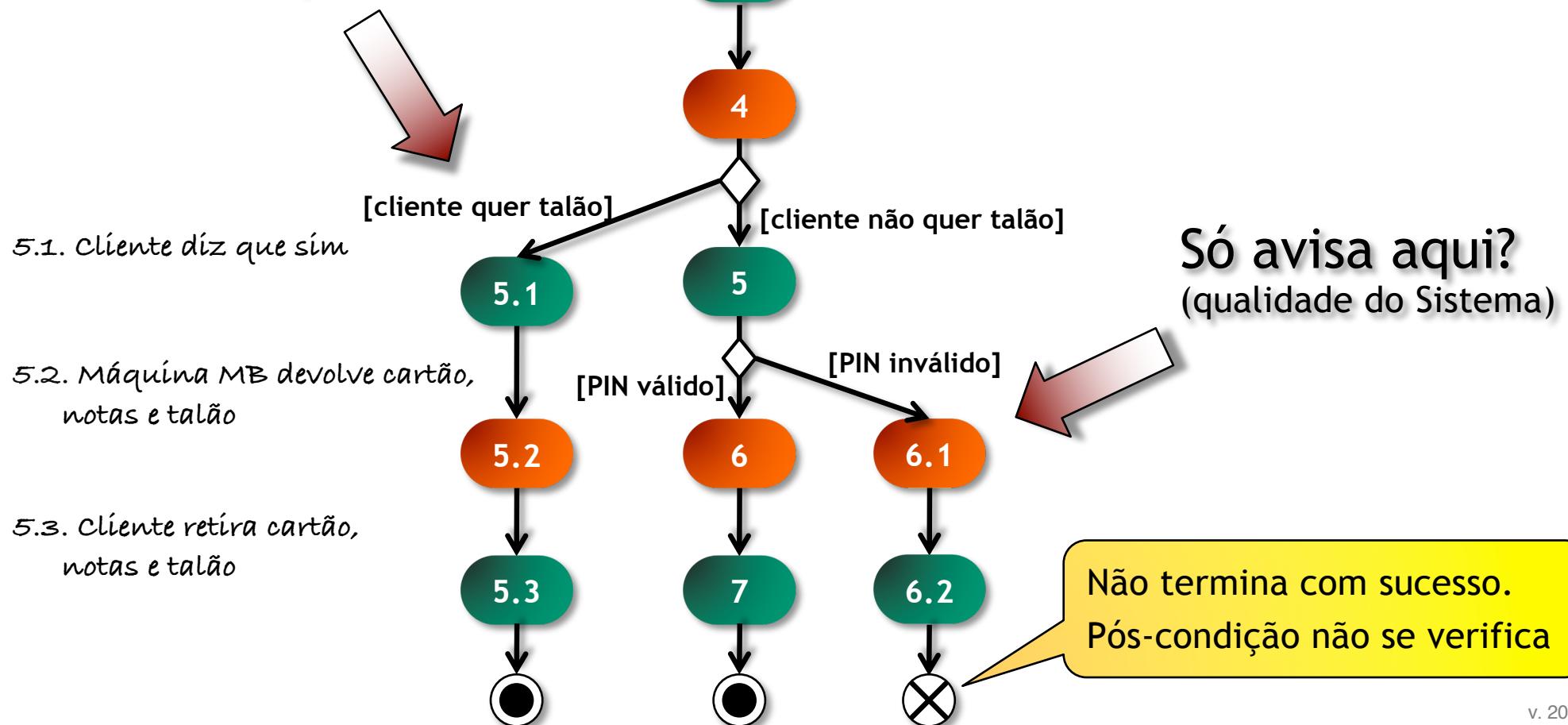
- **Fluxo de excepção 1:** [PIN inválido] (passo 6)

- 6.1. Máquina MB avisa sobre PIN inválido e fornece cartão
- 6.2. Cliente retira cartão



## Fluxo de exceção

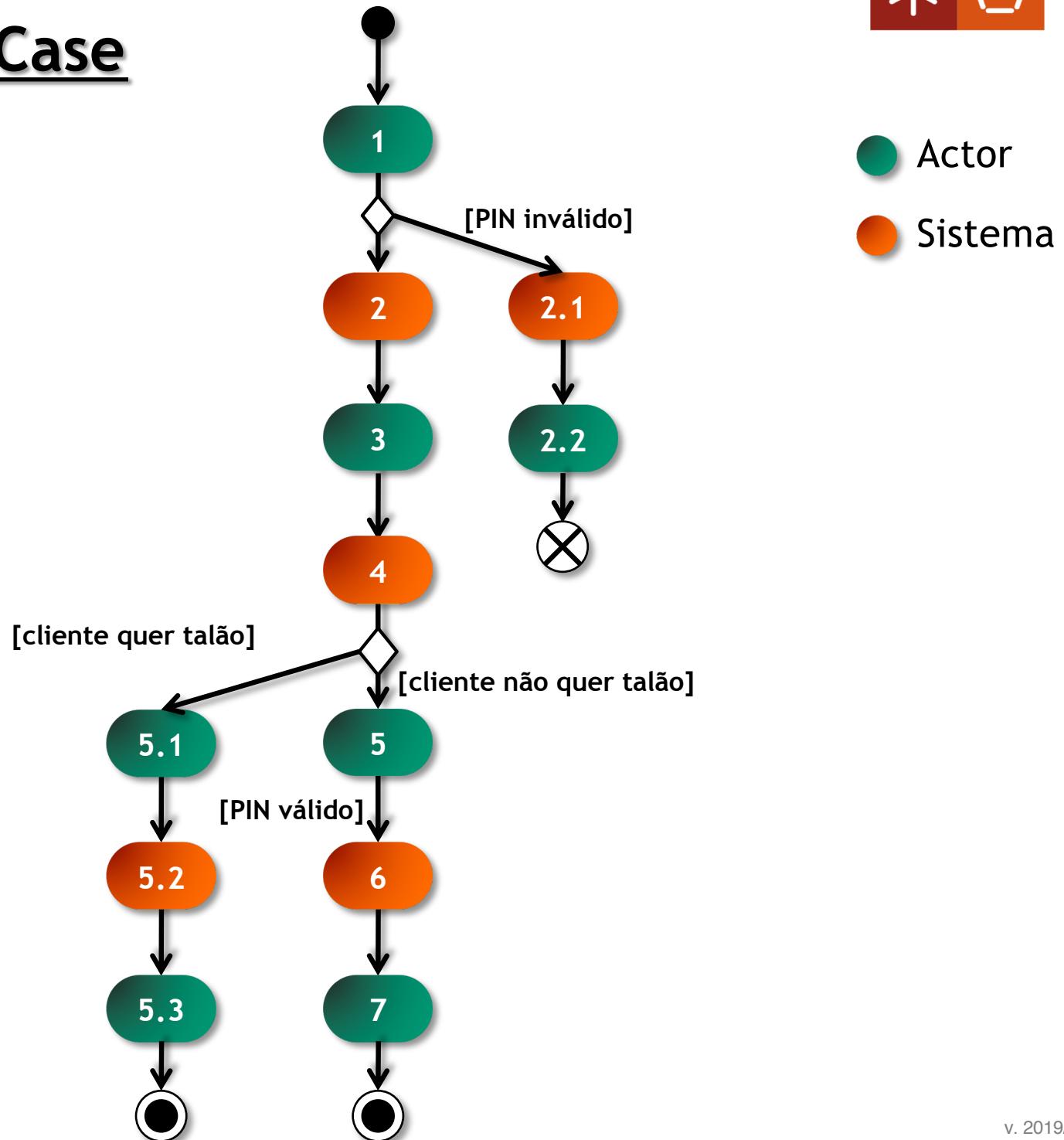
Quer talão e PIN inválido?  
(qualidade do *Use Case*)



# Exemplo - Máquina Multibanco

- **Use Case:** Levantar €
  - **Descrição:** Cliente levanta quantia da máquina
  - **Cenários:** O João levanta €60 com cartão
  - **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 e fornece notas
    7. Cliente retira cartão e notas
  - **Fluxo alternativo 1:** [Cliente quer talão] (passo 5)
    - 5.1. Cliente diz que sim
    - 5.2. Máquina MB devolve cartão, notas e talão
    - 5.3. Cliente retira cartão, notas e talão
  - **Fluxo de excepção 1:** [PIN inválido] (passo 2)
    - 2.1. Máquina MB avisa sobre PIN inválido e fornece cartão
    - 2.2. Cliente retira cartão

# Melhorar o Use Case



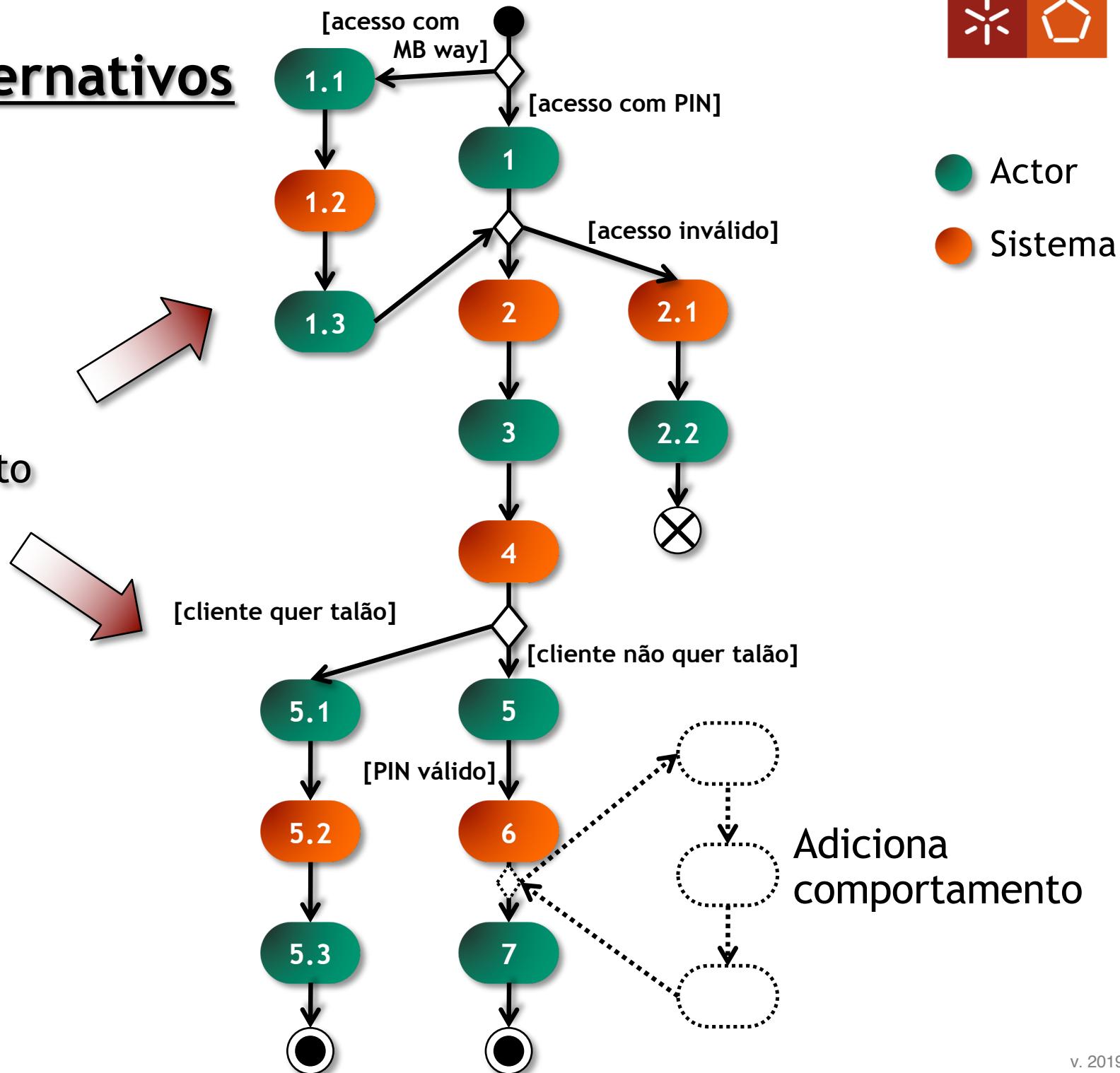
# Exemplo - Máquina Multibanco

- **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 e fornece notas
    7. Cliente retira cartão e notas
  - **Fluxo alternativo 1:** [Cliente quer talão] (passo 5)
  - ...
  - **Fluxo alternativo 2:** [Cliente autentica-se com MB way] (passo 1)
    - 5.1. Cliente escolhe acesso MB way
    - 5.2. Máquina MB pede Código MB way
    - 5.3. Cliente indica Código MB way
    - 5.4. Regressa a 2
  - **Fluxo de excepção 1:** [PIN inválido] (passo 2)
    - ...

MB way?! Temos que actualizar o Modelo de Domínio!!

# Fluxos alternativos

Substituem comportamento



Adiciona comportamento

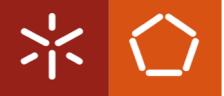
# Exemplo - Máquina Multibanco

- **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 e fornece notas
    7. Cliente retira cartão e notas
  - **Fluxo alternativo 1:** [Cliente quer talão] (passo 5)
    - 5.1. Cliente diz que sim
    - 5.2. Máquina MB devolve cartão, notas e talão
    - 5.3. Cliente retira cartão, notas e talão
  - **Fluxo alternativo 3:** [Cliente autentica-se com]
    - 5.1. Cliente prima escolhe acesso MB way
    - 5.2. Máquina MB pede Código MB way
    - 5.3. Cliente indica Código MB way
    - 5.4 Regressa a 2
  - **Fluxo de exceção 1:** [PIN inválido] (passo 2)
    - 2.1. Máquina MB avisa sobre PIN inválido e fornece
    - 2.2. Cliente retira cartão

- Cartão ilegível?
- € insuficiente na máquina?
- Valor diário excedido?
- Quantia impossível com notas existentes?
- Cliente quer desistir?
- Ligação ao servidor cai? (!)
- Dispensador de notas encrava? (!)
- etc., etc., etc.

# Use Cases - Especificação

- Não escrever *Use Cases* demasiados longos
  - Focar no que é essencial garantir
- Entidades referidas no Use Case devem estar presentes no Modelo de Domínio
  - Modelo de Domínio descreve o contexto do problema
  - Modelo de Use Case descreve uma solução
  - Conceitos têm que ser os mesmos!
- Deve ser expresso ao nível dos requisitos dos Actores (utilizadores/sistemas)
  - Não devem especificar a interface com o utilizador!!



# Modelação do Requisitos Funcionais

## Sumário:

- Requisitos funcionais vs. requisitos não funcionais
- Definição de requisitos funcionais
- Definição de Use Case
- Identificação de Use Cases e Actores
  - O papel dos cenários
- Representação textual de Use Cases