

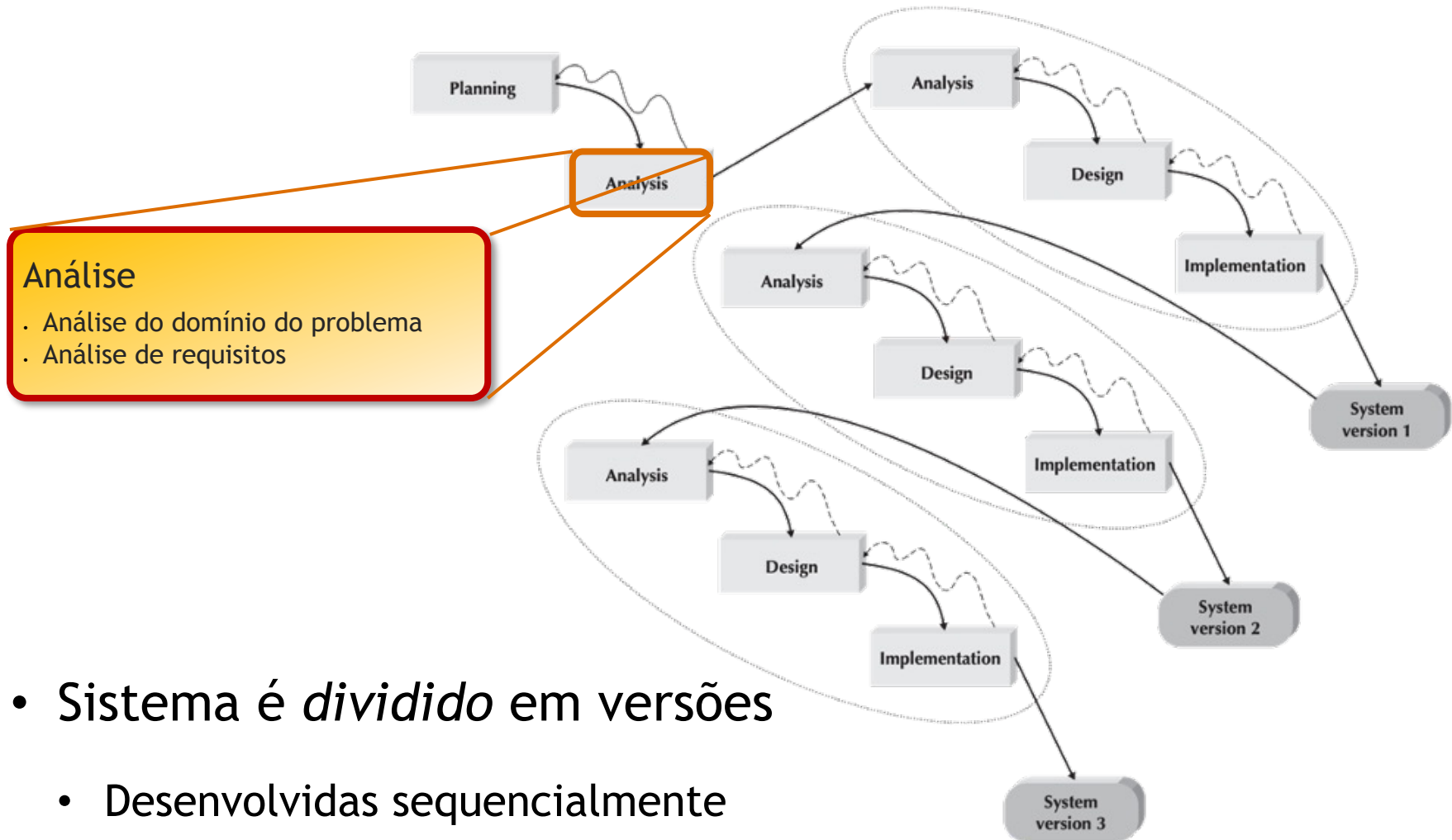


# Desenvolvimento de Sistemas de Software

## Modelação do Requisitos Funcionais (*Use Cases* - Casos de Uso)

# Desenvolvimento Iterativo e Incremental

- Desenvolvimento faseado (*Phased development*)



- Sistema é *dividido* em versões
  - Desenvolvidas sequencialmente



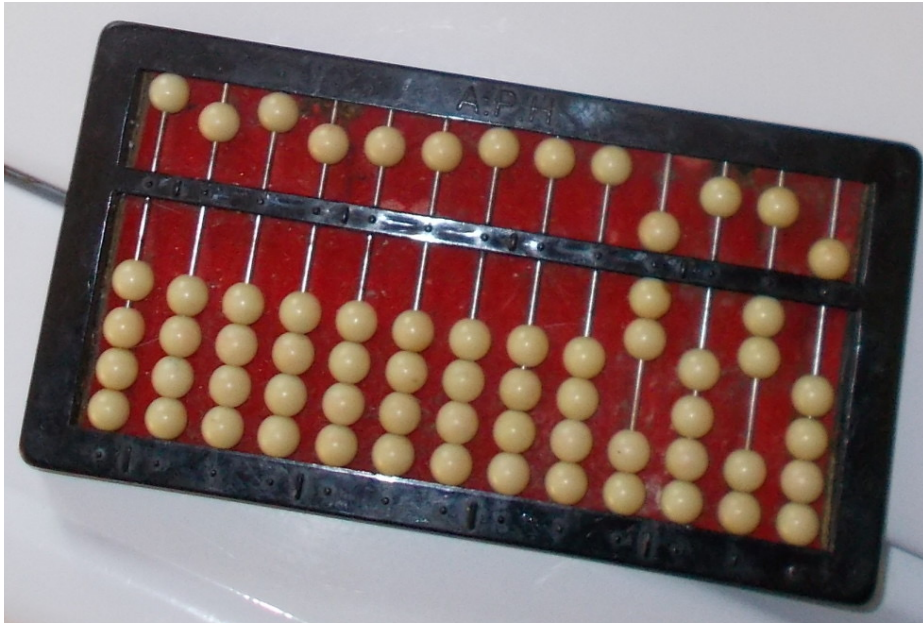
# Requisitos

- Os requisitos de software são as condições ou capacidades que o sistema deve satisfazer ou possuir para atender às necessidades dos utilizadores, clientes ou partes interessadas (*stakeholders*).
- Técnicas de Elicitação de Requisitos:
  - Entrevistas
  - Questionários
  - Workshops



# Tipos de requisitos

- Preciso de fazer *contas*...

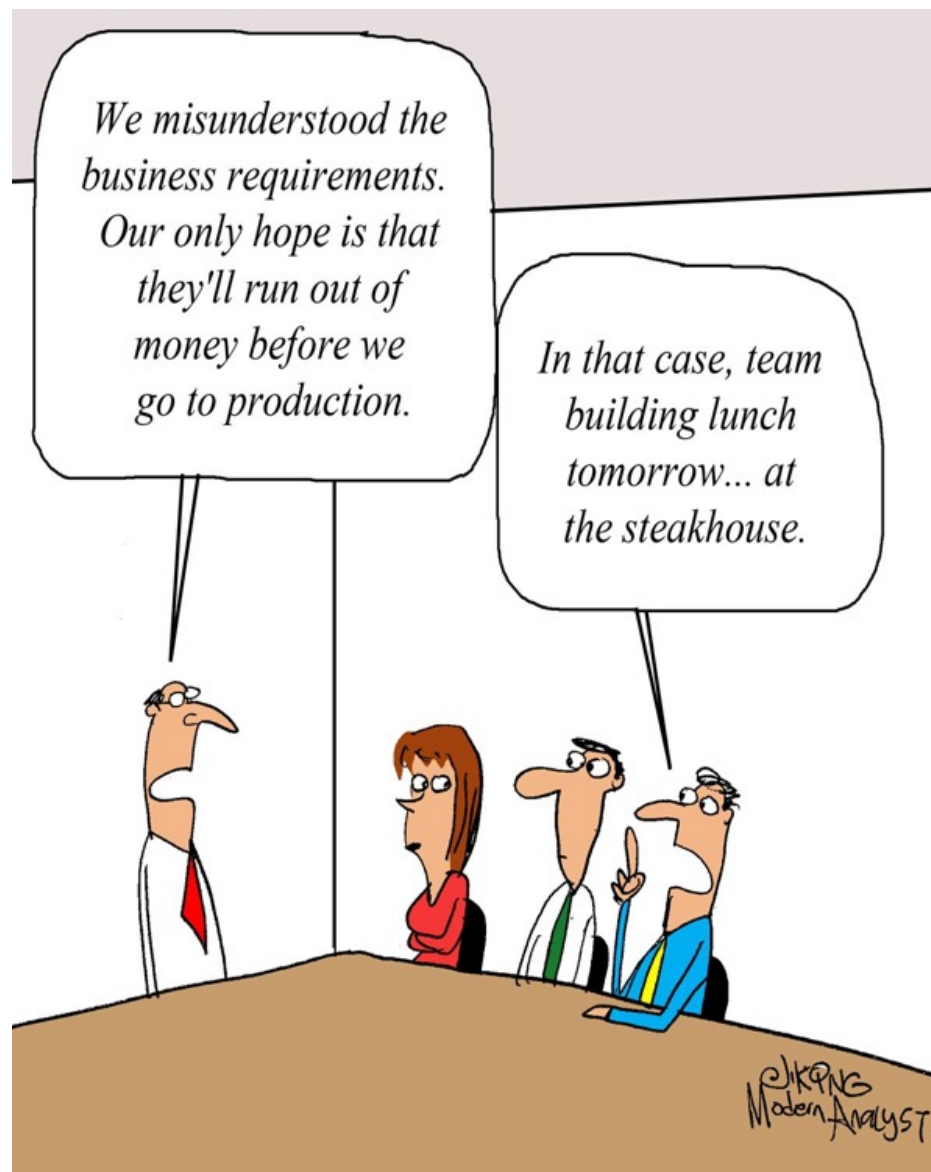




# Tipos de requisitos

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

# Importância dos requisitos





# Validação dos requisitos

- Os requisitos são validados com o cliente
- Especificação deve ser :
  - **Completa**  
*Todos os aspectos relevantes foram considerados*
  - **Consistente**  
*Não existem contradições entre requisitos*
  - **Inequívoca** (sem ambiguidades)  
*Nenhum requisitos pode ser interpretado de formas diferentes*
  - **Correcta**  
*Os requisitos descrevem correctamente o que o cliente pretende e o que a equipa de desenvolvimento se propõe fazer*



# Validação dos requisitos

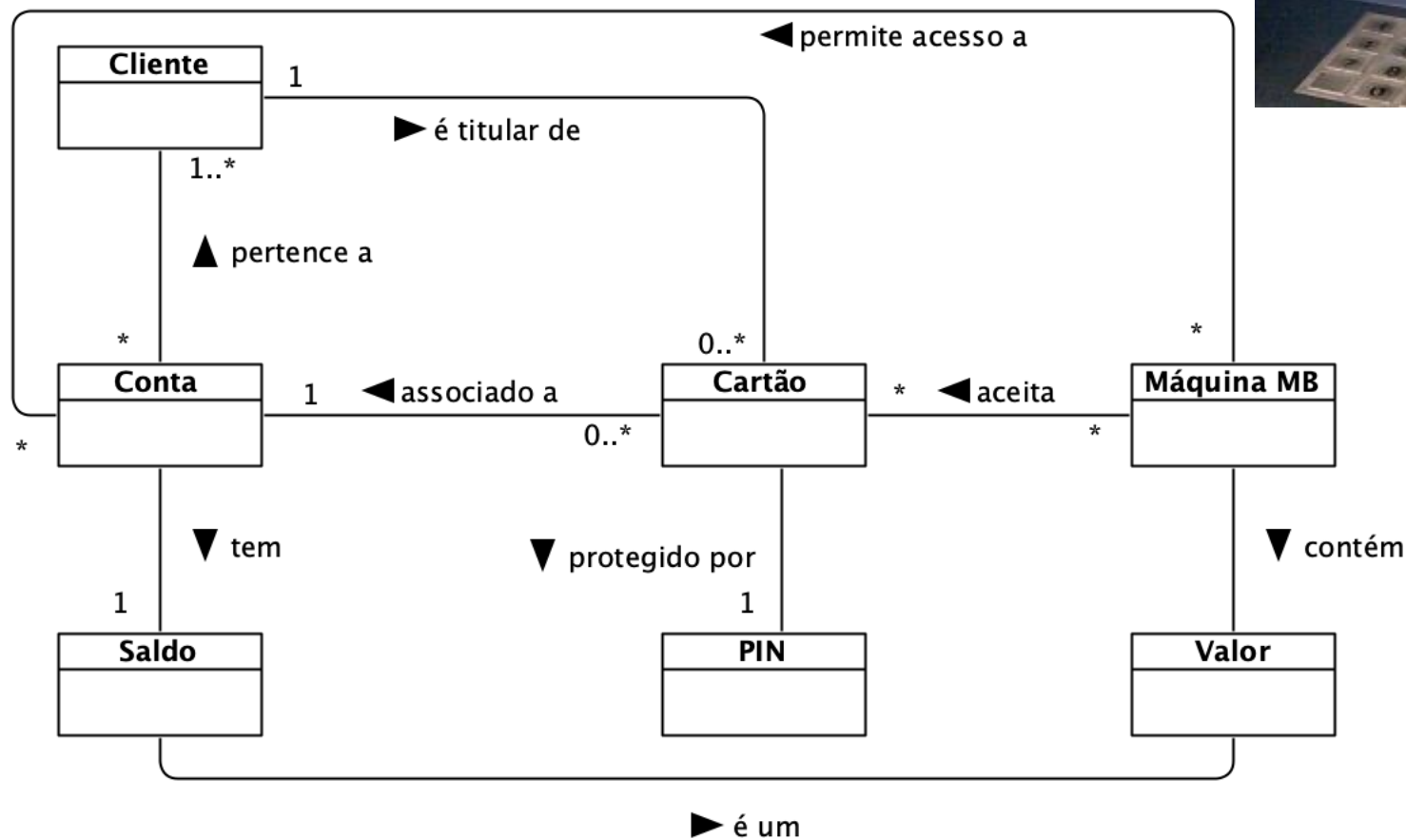
- A especificação de requisitos deve ainda ser...
  - **Realista**  
Não se deve prometer o que não podemos realizar
  - **Verificável**  
Temos que poder saber se cumprimos os objectivos!
  - **Rastreável**  
Temos que poder saber porque é que cada requisito foi definido





# 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. Introduce 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 cartão e o valor pedido, que o João retira.
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. ...

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





# 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. ...

## • Requisitos funcionais

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

## • Actores

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

Vamos especificar (descrever) os requisitos com *Use Cases* (Casos de Uso)

Sempre um verbo!  
*Fazer algo...*

Quem usa as funcionalidades do Sistema



# Identificação de requisitos / Casos de Uso

- Etapas a cumprir (com o auxílio de cenários de utilização do sistema):
  1. Identificar actores (quem vai utilizar o sistema)
  2. Identificar casos de uso (o que se vai poder fazer no sistema)
  3. Identificar associações (quem vai poder fazer o quê)
- **Identificar actores**
  - Um actor não é necessariamente um humano, pode ser um outro sistema, etc.
  - Cada actor representa um papel (“role”) que “alguém” ou alguma “coisa” externa ao sistema, e com interesse, nele pode assumir
  - Neste caso: Cliente, Bancário, Técnico de Manutenção?, Servidor do banco?
- **Identificar Casos de Uso**
  - Objectivos dos utilizadores/actores?
  - Resposta a estímulos externos.



# 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:

(cf. Verificável!)  
(cf. Inequívoca!)

**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 (o que deve ser verdade depois do use case)
- **Fluxo normal:**
  - Fluxo de eventos mais comum
- **Fluxos alternativos:**
  - Especificação dos modos alternativos de completar o use case
- **Fluxos de excepção:**
  - Especificação de situações relevantes em que o use case não tem sucesso

(cf. Rastreável!)

(cf. Completa!)



# Exemplo - Máquina Multibanco

## Use Case: *Levantar €*

**Descrição:** *Cliente levanta quantia da máquina*

**Cenários:** *Cenário 1 (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:**

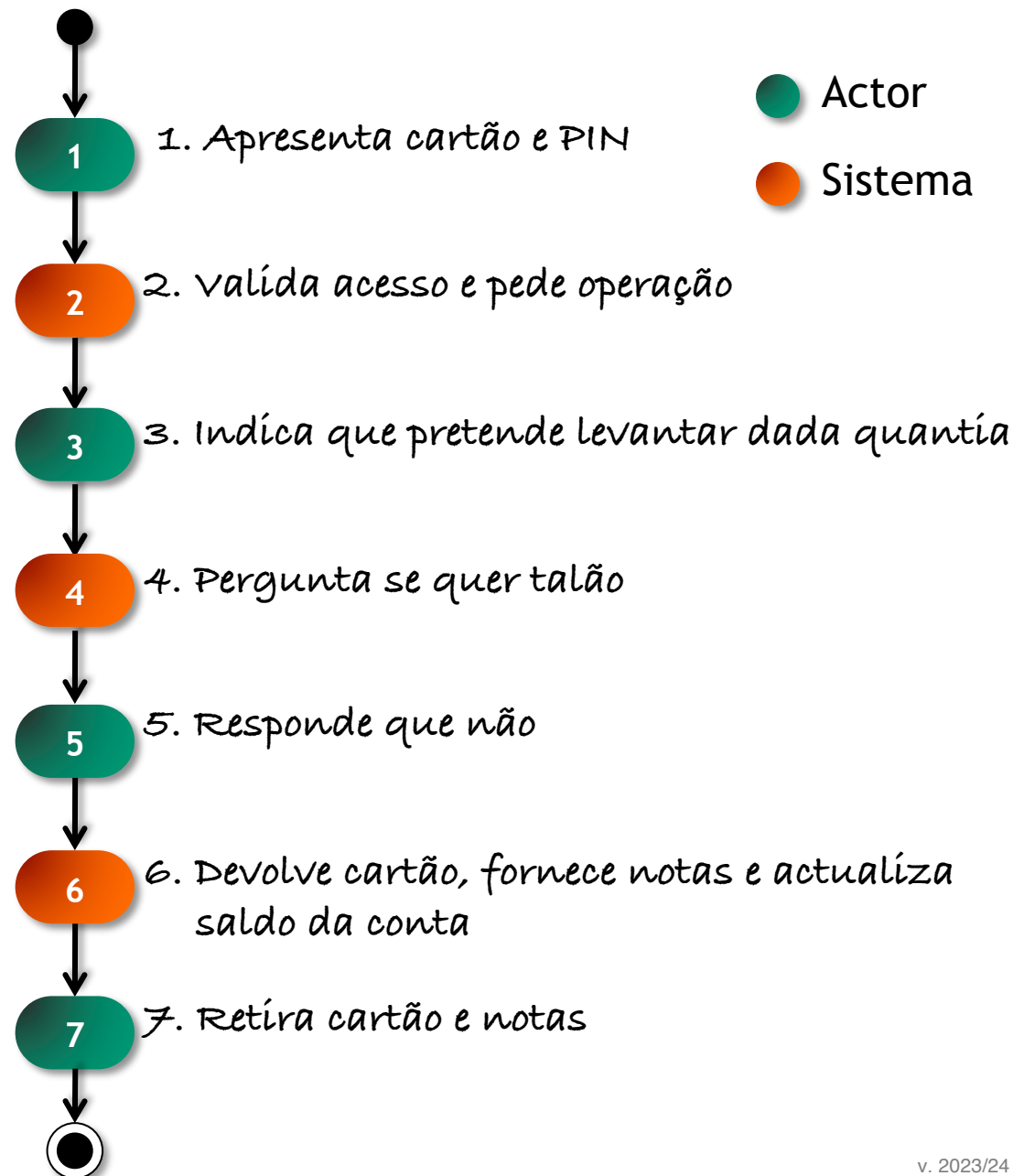
1. *Cliente apresenta cartão e PIN*
2. *Máquina MB pede operação*
3. *Cliente indica que pretende levantar dada quantia*
4. *Máquina MB pergunta se quer talão*
5. *Cliente responde que não*
6. *Máquina MB devolve cartão, fornece notas e actualiza saldo da conta*
7. *Cliente retira cartão e notas*

### ***O João levanta €60 com cartão***

O João vai viajar e dirige-se a uma máquina MB para levantar dinheiro para a viagem. Introduce 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 cartão e o valor pedido, que o João retira.



# Fluxo normal





# Exemplo - Máquina Multibanco

## Use Case: Levantar €

**Descrição:** Cliente levanta quantia da máquina

**Cenários:** Cenário 1 (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, fornece notas e actualiza saldo da conta~~
7. ~~Cliente retira cartão e notas~~

(cf. Completa!)

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

- 5.1. Cliente responde que sim ✓
- 5.2. Máquina MB devolve cartão, notas e talão e actualiza saldo da conta ✓
- 5.3. Cliente retira cartão, notas e talão ✓

### Fluxo de excepção (2): [PIN inválido] (passo 6)

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





# Fluxo alternativo





# Exemplo - Máquina Multibanco

## Use Case: Levantar €

**Descrição:** Cliente levanta quantia da máquina

**Cenários:** Cenário 1 (O João levanta €60 com cartão)

**Pré-condição:** Sistema tem notas

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

Não termina com sucesso.  
Pós-condição não se verifica

### 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, fornece notas e actualiza saldo da conta~~
7. ~~Cliente retira cartão e notas~~

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

- 5.1. Cliente responde que sim
- 5.2. Máquina MB devolve cartão, notas e talão e actualiza saldo da conta
- 5.3. Cliente retira cartão, notas e talão

### Fluxo de excepção (2): [PIN inválido] (passo 6)

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



# Exemplo - Máquina Multibanco

## Use Case: Levantar €

**Descrição:** Cliente levanta quantia da máquina

**Cenários:** Cenário 1 (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, fornece notas e actualiza saldo da conta
7. Cliente retira cartão e notas

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

- 5.1. Cliente responde que sim
- 5.2. Máquina MB devolve cartão, notas e talão e actualiza saldo da conta
- 5.3. Cliente retira cartão, notas e talão

### Fluxo de excepção (2): [PIN inválido] (passo 6)

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





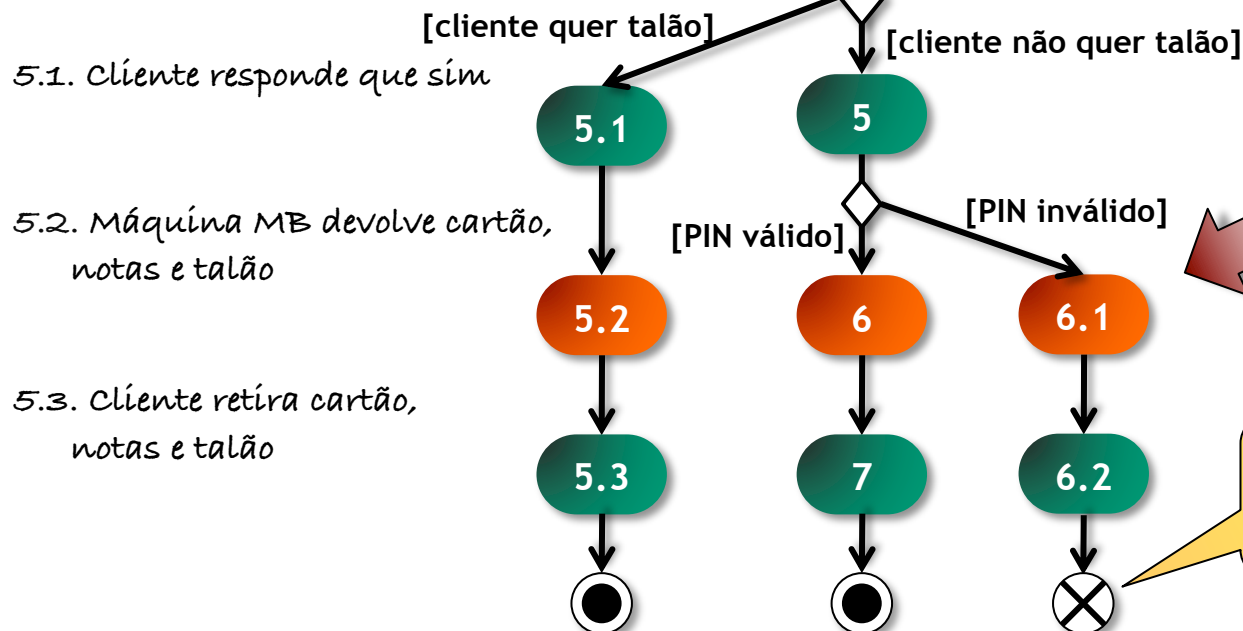
# Fluxo de excepção

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

(cf. Correcta!)

Algun problema?

● Actor  
● Sistema



Só avisa aqui?  
(qualidade do Sistema)

Não termina com sucesso.  
Pós-condição não se verifica



# Exemplo - Máquina Multibanco

## Use Case: Levantar €

**Descrição:** Cliente levanta quantia da máquina

**Cenários:** Cenário 1 (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 **válida 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 da conta
7. Cliente retira cartão e notas

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

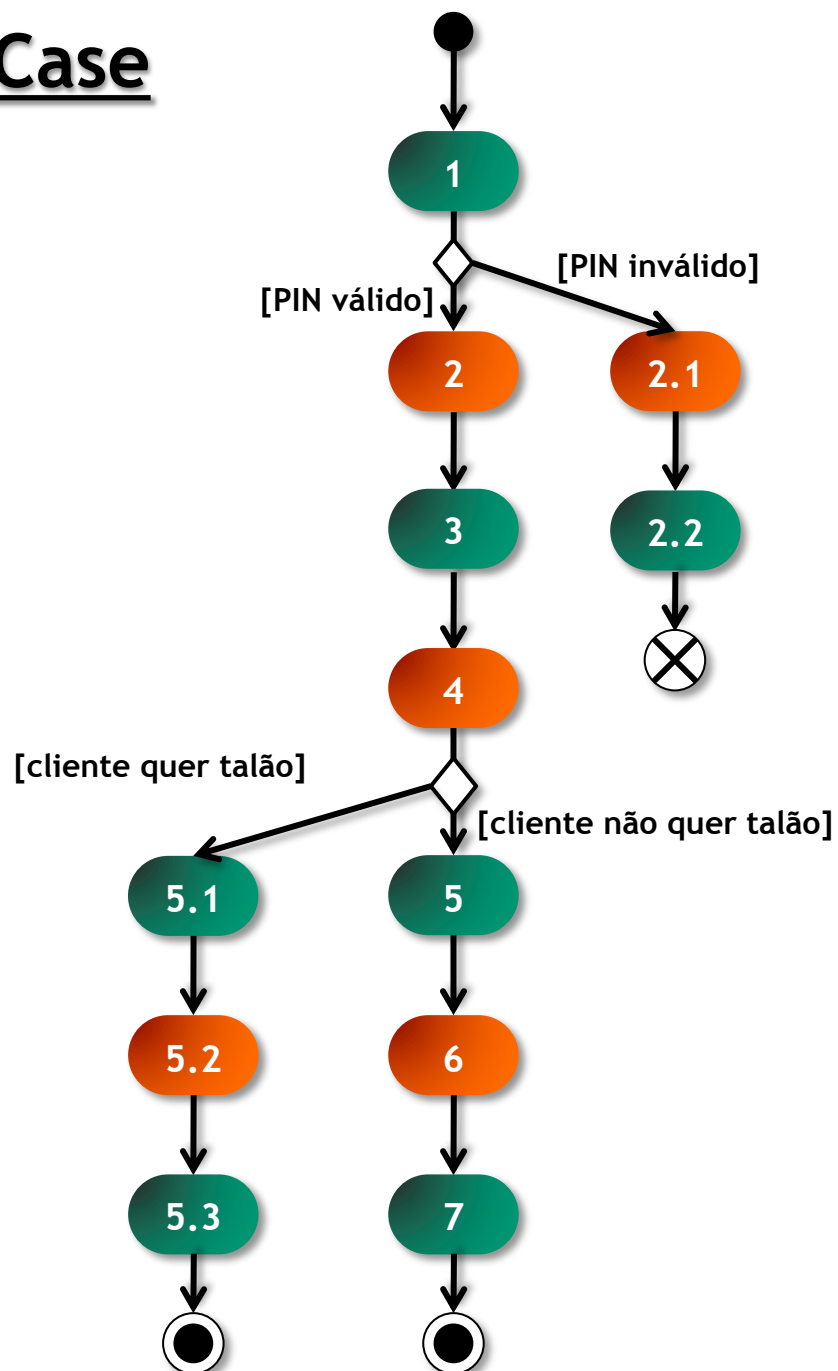
- 5.1. Cliente responde que sim
- 5.2. Máquina MB devolve cartão, notas e talão e actualiza saldo da conta
- 5.3. Cliente retira cartão, notas e talão

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

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



# Melhorar o Use Case



● Actor  
● Sistema



# Exemplo - Máquina Multibanco

## Use Case: Levantar €

**Descrição:** Cliente levanta quantia da máquina

**Cenários:** Cenários 1 e 2 (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 da conta
7. Cliente retira cartão e notas

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

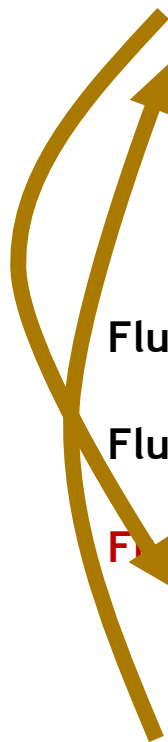
...

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

...

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

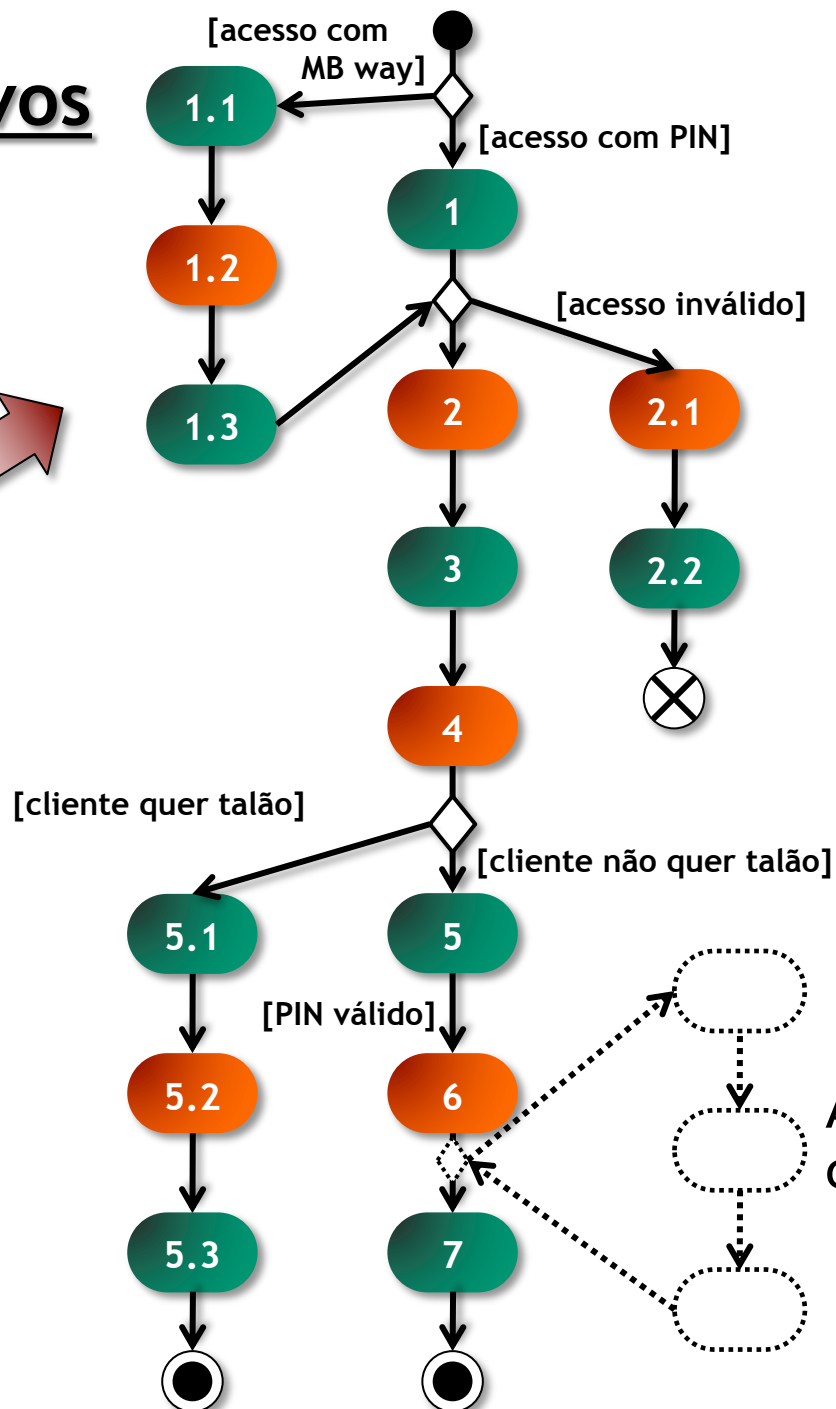
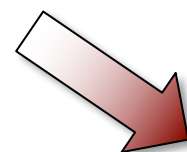
- 1.1. Cliente escolhe acesso MB way
- 1.2. Máquina MB pede Código MB way
- 1.3. Cliente indica Código MB way
- 1.4 Regressa a 2





# Fluxos alternativos

Substituem  
comportamento



Actor  
Sistema

Adiciona  
comportamento





# Exemplo - Máquina Multibanco

## Use Case: Levantar €

**Descrição:** Cliente levanta quantia da máquina

**Cenários:** Cenários 1 e 2 (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 da conta
7. Cliente retira cartão e notas

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

- 5.1. Cliente responde que sim
- 5.2. Máquina MB devolve cartão, notas e talão
- 5.3. Cliente retira cartão, notas e talão e actualiza saldo da conta

### 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 quer autenticar-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

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



# Processo de definição de requisitos (funcionais)

## Visão orientada aos *Use Case*

1. Recolher **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 compreensíveis para os clientes
2. Identificar e especificar os *Use Cases* (Casos de Uso) - descrições de fluxos de interacção com o Sistema por forma a suportar os cenários
3. Identificar **Actores** - entidades externas que interagem com o sistema
4. Identificar relações entre Actores e *Use Cases*

## *Vantagens:*

- Não há trabalho desnecessário.
- O Sistema de Software 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*)



# Use Cases - Especificação (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.



# 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!!



# *Use Cases*

*“Good use cases are balanced, describing essential system behavior while providing only the necessary details about the interactions between system and its users”*

(S. Adolph & P. Bramble - *Patterns for Effective Use Cases*.  
The Agile Software Development Series. Addison-Wesley Professional)