

# Anotações de teoria - Use cases

3 de outubro de 2023 23:23

## 1 Tipos de requisitos:

Requisitos funcionais - O que o sistema deve fazer

- descrevem as interações entre o Sistema e o seu ambiente
- Ambiente: utilizadores e outros sistemas
- Descrições independentes da implementação

Requisitos não funcionais - Como o sistema deve fazer-lo

- aspetos do sistema não diretamente relacionados com o seu comportamento funcional
  - o Usability, Dependability, Performance; e ainda: de implementação, de interface (com outros sistemas), de operação, de modo de entrega, legais

## 2 Validação dos requisitos

- os requisitos são validados com o cliente
- a especificação deve ser:
  - o completa - todos os aspetos relevantes foram considerados
  - o consistente - não existem contradições entre requisitos
  - o inequívoca - sem ambiguidades
  - o correta - os requisitos descrevem corretamente o que o cliente pretende e o que a equipa de desenvolvimento se propõe fazer
  - o realista - não se deve prometer o que não se pode realizar
  - o verificável - temos que poder saber se cumprimos os objetivos
  - o rastreável - temos que poder saber porque é que cada requisito foi definido

## 3 Identificação de requisitos / Casos de Uso

- Etapas a cumprir (com auxílio de cenários de utilização do sistema)
  1. Identificar atores (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 atores
  - o Um ator não é necessariamente um humano, pode ser um outro sistema, etc.
  - o Cada ator representa um papel que alguém ou alguma coisa externa ao sistema nele pode assumir
- Identificar casos de uso
  - o Objetivos dos utilizadores/ atores?
  - o Resposta a estímulos externos

## 4 Notação semiestruturada

Use Case: nome do use case

Descrição: breve descrição do use case

Cenários: cenários que originaram 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 do evento mais comum

Fluxos alternativos:

Especificação dos modos alternativos de completar o use case

Fluxos de exceção:

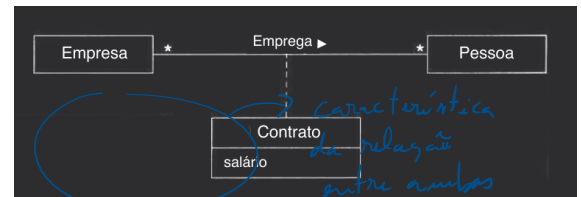
Especificação de situações relevantes em que o use case não tem sucesso

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. ...

Breves anotações sobre modelos de domínio:

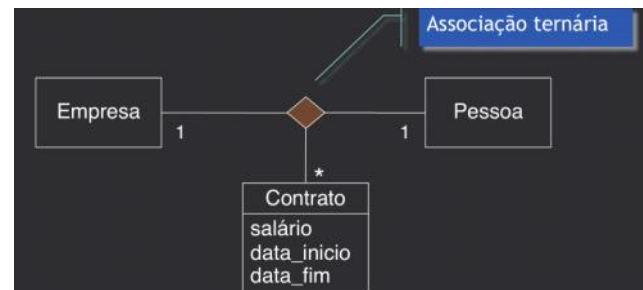
- Classes de associação:



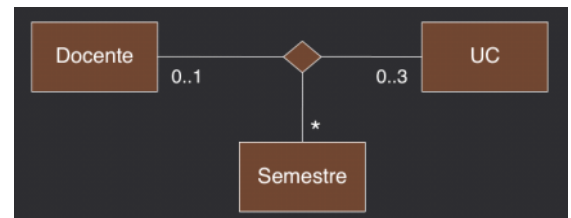
- o A relação entre cada empresa e cada um dos seus funcionários é caracterizada por um contrato
- o Cada Pessoa, pode ter estado contratada por várias empresas e para cada uma há um contrato diferente
- o O contrato não é característica da Empresa, nem da pessoa, mas da relação entre ambas

mas ... dois contratos diferentes com a mesma empresa?!

- Associações n-árias



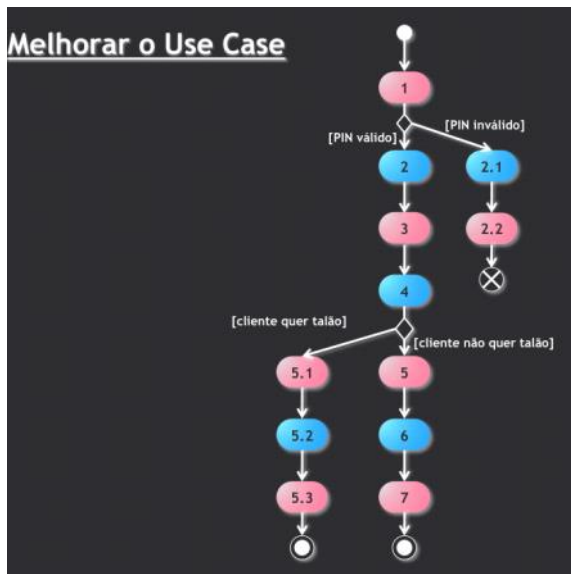
- o Multiplicidades indicam quantos objetos existem para uma dada combinação de objetos das outras classes
- o Navegabilidade, agregação e qualificação não são permitidos



- o Cada docente pode lecionar num semestre, no máximo, 3 Ucs
- o Uma multiplicidade de zero invalida a combinação de objetos!
  - Não é possível ter uma associação entre uma UC e um semestre sem indicar o docente, etc.

- Requisitos funcionais
  - Levantar €
  - Pagar serviço
  - Efectuar transferência
  - Carregar máquina
- Actores
  - Cliente (cf. João, Maria, Rui)
  - Funcionária (cf. Joana)

## Melhorar o Use Case



### 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)

...

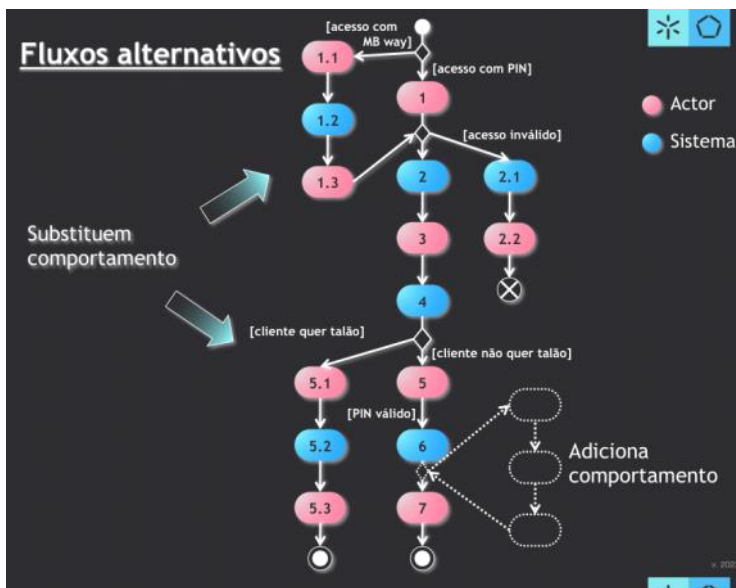
Fluxo de excepção (2): [PIN inválido] (passo 2)

...

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



Mais possibilidades:

- 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.

#### 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*
  - o Descreve como os atores atingem objetivos (realizam os use cases) utilizando o sistema
    - Definem relação entre inputs dos atores e comportamento do sistema
  - o Especificação deve incluir o comportamento tipicamente esperado, bem como variantes:
    - comportamentos alternativos que ainda levam ao sucesso
    - comportamentos de insucesso (exceções)
- Use cases - Especificação (Tipos de fluxos)
  - o Em cada especificação de um Use Case podem/devem existir diferentes fluxos de controlo (sequências de eventos, comportamentos)
  - o Podemos caracterizá-los em três tipos:
    - Fluxo normal (ou principal): Situação perfeita em que nada corre mal. A pós-condição é satisfeita no final (se pré-condição também o é no início)
    - Fluxo alternativo: fluxos válidos mas menos comuns. A pós-condição é satisfeita (se a pré-condição também o é no início)
    - Fluxos de exceção: condições de erro suficientemente importantes para serem capturadas no modelo. A pós-condição não é satisfeita.
  - o Não escrever Use Cases demasiado longos
    - Focar no que é essencial garantir
  - o Entidades referidas no Use Case devem estar presentes no Modelo de Domínio
    - Modelos de Domínio descreve o contexto do problema
    - Modelos de Use Case descreve uma solução
    - Conceitos têm de ser os mesmos!
  - o Deve ser expresso ao nível dos requisitos dos atores (utilizadores/ sistemas)
    - Não devem especificar a interface com o utilizador !!

*"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)