Use Cases

Metodologias e Desenvolvimento de Software

Pedro Salgueiro
pds@di.uevora.pt
CLV-256

UML - use cases

- **Engenharia de requisitos**
  - deve responder à perguntas:
    - O sistema a desenvolver **serve para quê**?
    - Deve **fazer o que?**
  - Interessa definir que “funcionalidades” o sistema deve exibir
    - para quem...
  - e não como se implementa

UML - use cases

- Use Cases
  - casos de utilização
  - permitem captar e ilustrar os requisitos do sistema
  - são uma representação/linguagem comum com os utilizadores e especialistas do domínio (negócio)
  - são cenários de utilização do sistema
    - os cenários são instâncias dos use cases

UML - use cases

- Não existem sistemas “isolados”
  - traçam as fronteiras do sistema com o “exterior”
  - especificam como esse “exterior” interage com o sistema a desenvolver
  - pode haver interacção com humanos ou com outros sistemas
  - são “diagramas de contexto”

UML - use cases

- “ingredientes” básicos
  - **Actores**
    - quem(ou o que) interage com o sistema
  - **Use Cases** (propriamente ditos)
    - Funcionalidades oferecidas aos actores

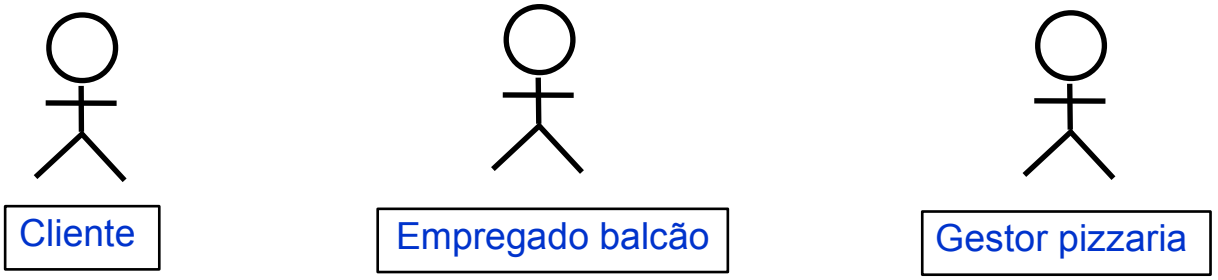
Exemplo

“Pretende-se desenvolver um sistema de informação de gestão para um grupo de pizzarias que permita aos clientes efectuar encomendas na loja ou através da Internet. Na loja, o cliente dirige-se ao empregado de balcão que introduz no sistema a encomenda do cliente.

“Caso a encomenda seja feita através da Internet o cliente terá que se identificar com um nome e palavra chave (controlo de acesso), e poderá usufruir de um desconto no item, caso esteja em promoção; portanto o cliente deve neste caso ter-se registado anteriormente. O sistema deve ainda permitir que o gestor da pizzaria efetue reservas de mesa, verificando se este tem autorização para o efectuar. O mesmo deverá acontecer para os restantes empregados.”

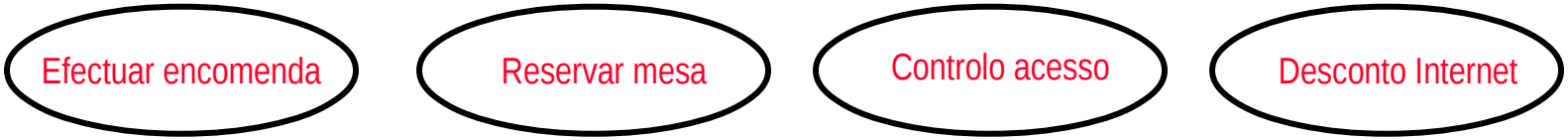
UML - use cases

- Actores
  - **não fazem parte do sistema**
    - mas são representados na sistema
      - base de dados
    - mas são cruciais para a análise



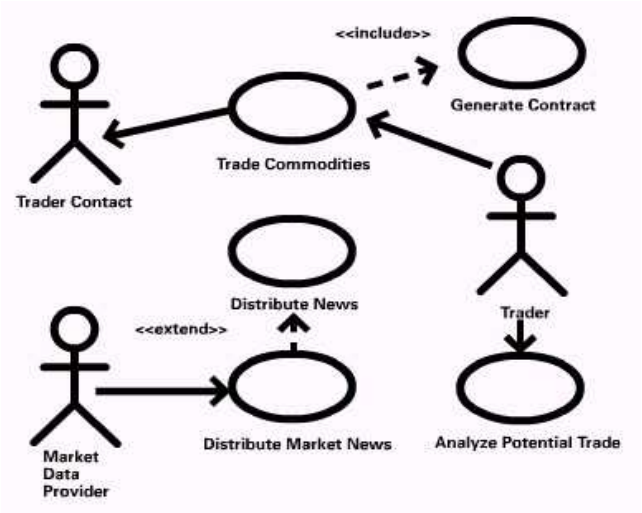
UML - use cases

- Use Cases
  - descrevem as grandes funcionalidades dos actores
  - devem "dar" um **resultado observável e útil para um actor;**
  - um **use case** especifica o comportamento do sistema (ou parte do sistema)
  - traduz-se num conjunto de **sequências de acções**
    - **incluindo variantes**



UML - use cases

- Associações
  - os **actores** ligam-se aos **use cases** por associação
  - uma associação entre um actor e um use case indica que há comunicação entre eles, com a possibilidade de envio/recepção de mensagens



UML - use cases

- um **use case <u>especifica o comportamento</u>** do sistema (ou parte do sistema)
  - não há regras absolutas!
  - vai-se ao nível de pormenor que permita descrever bem o problema;
  - pensar num sistema ou em sub-sistemas depende do analista e da dimensão do sistema;
  - *guide-line*
    - um use case deverá corresponder a uma grande funcionalidade bem definida que dá “um resultado” (a um capítulo de um hipotético manual de utilizador?);

UML - use cases

- Guide-line
  - numa ATM **não** interessa dizer que uma **"grande funcionalidade bem definida"** é:
    - "usar a ATM"!
  - mas também **não** que uma **"grande funcionalidade bem definida"** é:
    - "carregar na tecla dos € 20"
  - se calhar **poderia ser**:
    - levantar dinheiro
    - pagar uma conta
    - ver o saldo
    - activar a Via Verde
    - etc...

Exemplo

"Pretende-se desenvolver um sistema de informação de gestão para um grupo de pizzarias que permita aos clientes efectuar encomendas na loja ou através da Internet. Na loja, o cliente dirige-se ao empregado de balcão que introduz no sistema a encomenda do cliente.

Caso a encomenda seja feita através da Internet o cliente terá que se identificar com um nome e palavra chave (controlo de acesso), e poderá usufruir de um desconto no item, caso esteja em promoção; portanto o cliente deve neste caso ter-se registado anteriormente. O sistema deve ainda permitir que o gestor da pizzaria efetue reservas de mesa, verificando se este tem autorização para o efectuar. O mesmo deverá acontecer para os restantes empregados."

Exemplo

"Pretende-se desenvolver um sistema de informação de gestão para um grupo de pizzarias que permita aos clientes efectuar encomendas na loja ou através da Internet. Na loja, o cliente dirige-se ao empregado de balcão que introduz no sistema a encomenda do cliente.

Caso a encomenda seja feita através da Internet o cliente terá que se identificar com um nome e palavra chave (controlo de acesso), e poderá usufruir de um desconto no item, caso esteja em promoção; portanto o cliente deve neste caso ter-se registado anteriormente. O sistema deve ainda permitir que o gestor da pizzaria efetue reservas de mesa, verificando se este tem autorização para o efectuar. O mesmo deverá acontecer para os restantes empregados."

Exemplo

Actor	Use Case
Cliente	Efectuar encomenda Internet Desconto Controlo de acesso
Empregado balcão	Efectuar encomenda Controlo acesso
Gestor pizzaria	Reservar mesa Controlo acesso

UML - use cases

- Use case é composto por
  - **sequências de acções, incluindo variantes**
    - a sequência de acções pode ser descrita em **“texto corrido”** ou descrição **“estruturada”**;
    - deve começar-se pelo funcionamento **“normal”/base**, isto é, quando tudo corre bem;
    - mas depois há que descrever as **variantes**, por exemplo menos vulgares, de excepção, de erro;
    - interessa desde logo ver **conjuntos de acções comuns** a vários use cases; isolá-lo num use case!
      - ex: controlo de acesso

Exemplo

- funcionamento “normal”/base (texto corrido)

1) “O cliente, após ter validado o seu acesso ao sistema, selecciona a opção Encomendar, sendo então mostrado em simultâneo o catálogo de pizzas e a encomeda. Para adicionar um produto há apenas que introduzir o código do mesmo para que apareça, automaticamente, a sua descrição, o preço e o total.”

2) “Através da opção Confirmar é confirmada a encomenda e passa-se para o pagamento onde, após a introdução e confirmação dos dados do cartão de crédito é atribuído um número de identificação à encomenda que será entregue na morada do cliente.”

Exemplo

- funcionamento “normal”/base (estruturado)
  1) o cliente valida-se perante o sistema;
  2) o use case começa quando o cliente selecciona a opção Encomendar;
  3) o sistema mostra a encomenda e simultaneamente o catálogo de pizzas;
  4) o cliente adiciona produtos à encomenda através do código;
  5) o sistema “completa” com a descrição e preço do produto;
  6) o sistema mostra também o total;
  7) o cliente confirma a encomenda com a opção Confirmar;
  8) o sistema pede elementos do cartão de crédito;
  9) o cliente dá esses elementos ao sistema;
  10) o sistema confirma os dados e indica um número de encomenda;

Exemplo

- **variantes**
  - vimos o cenário principal, em que “tudo corre bem”
  - o que é que pode variar?
    - por exemplo: código de produto inexistente, cartão de crédito não válido
  - **4**. o cliente adiciona produtos à encomenda através do código
    - **4a**. se o código não existe o sistema avisa e volta a 3.
  - **10**. o sistema confirma os dados e indica um número de encomenda.
    - **10a**. se o cartão for inválido o sistema avisa o cliente e volta ao passo 8
  - a qualquer momento o cliente pode desistir da encomenda usando a opção Cancelar

Acções comuns

1) “*o cliente valida-se perante o sistema*”

- provavelmente a **“validação perante o sistema”** será comum a vários use cases;
- é um procedimento reutilizável noutros casos ou que já existia no sistema
- recorro então à figura de **<<uses>> Validação** perante o sistema

Extensões

5) "*o sistema 'completa' com a descrição e preço do produto'*

- **5x)** se o produto está em promoção com desconto
  - ***Calcular desconto***

- é um "sub-use case" que é usado apenas em certas condições
  - ele próprio tem uma descrição
    - 1. o sistema retorna o valor do desconto que mostra ao cliente
    - 2. calcula o desconto subtraindo ao preço do produto

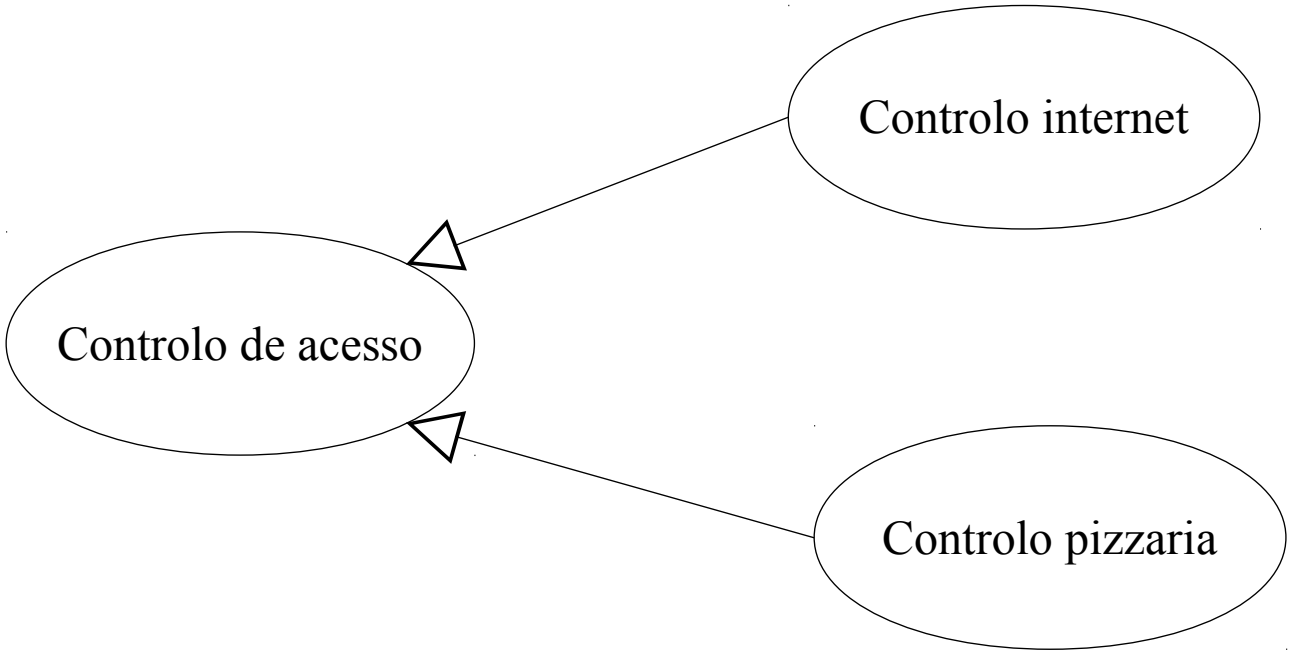
- recorro à figura de **<<extends>> Calcular desconto**

generalizações/especializações

- herança
- aplicável a **use cases** e a **actores;**
- são **particularizações**
  - com um **tronco comum**, o use case ou actor geral;

UML - use cases

- generalizações/especializações
  - A generalização possui as mesmas propriedades duma relação pai/filho, onde o **use case** "filho" herda ou substitui por completo o comportamento do *use case* "pai":



Herança entre actores

UML - use cases

- Resumindo
  - um **use case** envolve a **interacção** entre um actor e o sistema;
  - um **actor** representa um tipo de **“papel” (role)** coerente que interage com o use case;
  - a interacção deve referir a perspectiva do ator mas também a resposta do sistema;
  - deve descrever o “ping-pong” actor**-sistema;**
  - um use case pode ter **variantes;**
  - pode haver use cases (actores) que são especializações de outros use cases (actores) gerais (“herança”);
  - use cases que usam ou incluem (**uses/includes**) outros use cases;
  - use cases que estendem ou acrescentam **(extends)** outros use cases base (“tratamento de situações especiais”);

UML - use cases

- Outras utilizações
  - ajudam a conceber e validar a arquitectura do sistema;
    - a estrutura dos use cases é determinante;
  - ao longo da implementação **concretizam-se** por **colaborações** entre elementos;
  - são a base para **testes** do sistema;
  - são um “ponto de referência” para todas as facetas do desenvolvimento;

UML - use cases

- importante
  - o use case descreve um sistema a um nível (muito) elevado de **abstracção**;
  - não se diz como serão implementados ;
  - é uma disciplina de análise difícil;
  - é fácil resvalar para a tecnologia;
  - é fácil descer de nível de abstracção;
  - cuidado com as **especializações** e os **extends**

UML - use cases

- importante
  - **não se diz como serão implementados...**
    - por exemplo um sistema de táxi digital, na perspectiva de use case, fala em localização remota de veículos, mensagens de pré-alarme e alarme, de actuação sobre o veículo
    - mas não é necessário **(não deve)** referir como funciona internamente;
      - nunca referir GPS, GSM, SMS, trunking, rádio

UML - use cases

- como identificar actores?
  - quem vai beneficiar com o sistema?
  - quem está interessado em certo requisito do sistema?
  - onde, na organização, é que o sistema é usado?
  - quem vai fornecer a informação ao sistema, quem vai usá-la, quem vai retirá-la?
  - o sistema usa algum recurso externo?
  - existe alguém que tenha diferentes papéis (responsabilidades)?
  - existem diversas pessoas com o mesmo papel? ou parecidos?

UML - use cases

- conteúdo típico de um use case

  - como se despoleta o use case;
  - onde começa e acaba o use case;
  - pontos de interacção com os actores;
  - que dados são trocados;
  - fluxo de eventos normal;
  - fluxos de eventos alternativos;
  - fluxo de eventos excepcional;

uses & extends

- The UML User Guide (a Bíblia)
  - Um **include** (uses)... significa que o use case base, explicitamente incorpora o comportamento de outro use case...
  - Usa-se um **include**... para evitar descrever o mesmo fluxo de eventos várias vezes, **colocando o comportamento comum** num use case.

UML – uses & extends

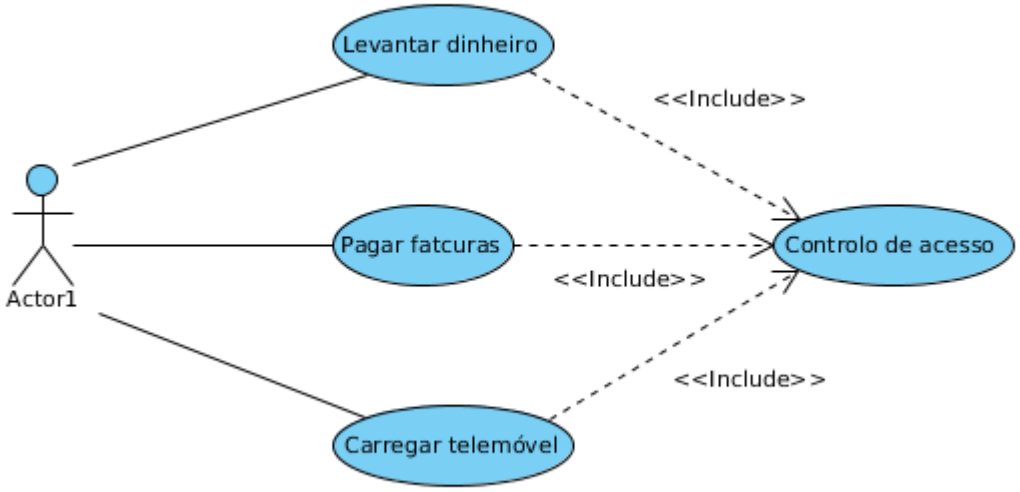
- The UML User Guide (cont.)
  - Um extends… significa que o use case base incorpora implicitamente o comportamente de outro use case...
  - Usa-se um extends... para modelar parte de um use case, que o utilizador pode ver como um comportamento opcional. Desta forma, separa-se o comportamento opcional do comportamento “normal”.
  - Pode-se usar um extends... para modelar um fluxo separado que é executado apenas sob certas condições.
  - Por fim, usa-se um extends... para modelar vários fluxos que podem ser inseridos em vários pontos, geridos explicitamente pela interação com um ator.

UML – uses & extends

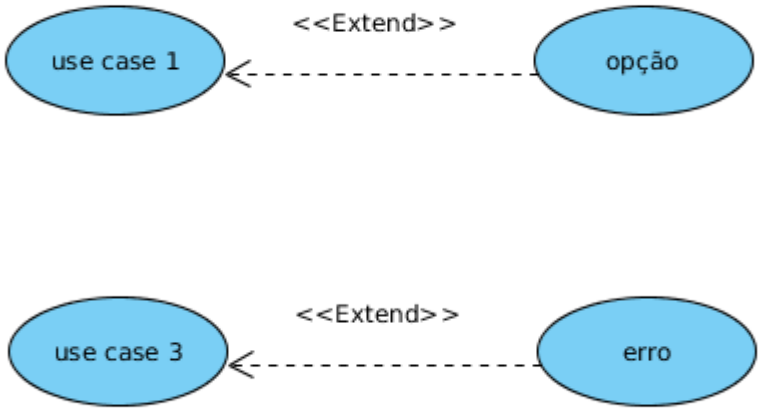
- The UML User Guide (cont.)
  - Organizar os use cases, extraindo:
    - Comportamento comum (include/uses) **e**
    - Variantes ou excepções (extends)
  - Importante na criação de um conjunto de use cases simples, equilibrado e de fácil compreensão.

UML – uses & extends

common behaviour (<<include>> / <<uses>>)

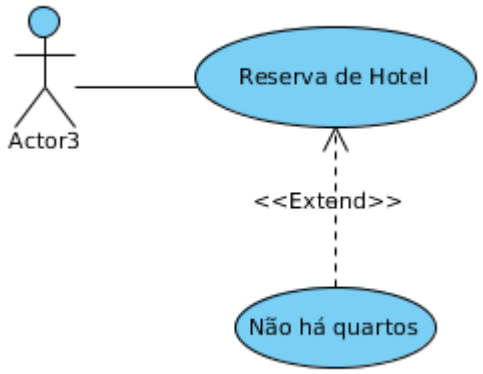


distinguishing variants (<<extends>>)



UML – uses & extends

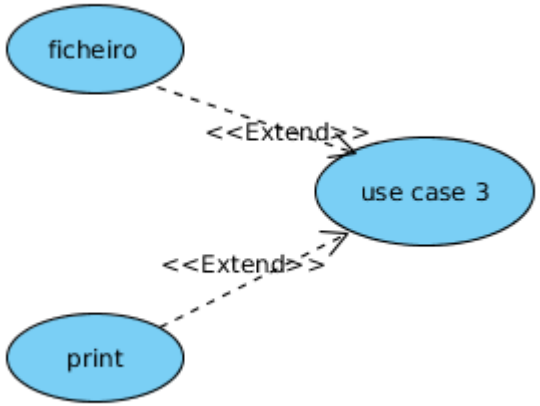
- Nome: Reserva de hotel
- Quem inicia: Cliente
- Objectivo: Reservar um quarto num hotel



- Cenário Principal
  - 1. O Cliente pede para fazer uma reserva
  - 2. O Cliente selecciona o hotel, as datas, o tipo de quarto
  - 3. O Sistema fornece disponibilidades e preço
  - 4. O Cliente decide uma escolha
  - 5. O Cliente fornece a sua identificação e contacto
  - 6. O Cliente fornece pormenores sobre o modo de pagamento
  - 7. O Sistema faz a reserva e dá uma referência
  - 8. O Sistema envia uma confirmação de reserva

- Extensão
  - 3. Não há quartos disponíveis
    - a-O Sistema fornece outros tipos de quartos ou outro hotel
    - b-O Cliente selecciona uma alternativa

distinguishing variants (<<extends>>)



- Cenário Principal
  - 4. O Cliente confirma a encomenda
  - 5. O Sistema passa um recibo
  - ...
- Extensão
  - 5. O Sistema pergunta pergunta ao utilizador:
    - a. Impressão do recibo
    - b. Recibo guardado em ficheiro