

Use Cases

Metodologias e Desenvolvimento de Software
2022/2023

Pedro Salgueiro
CLAV-256
pds@uevora.pt

Use Cases

O que são?

- Técnica para capturar requisitos funcionais de um sistema
- Descrevem interações entre:
 - Utilizadores do sistema
 - Sistema;
- Mostram como o sistema é usado;

Cenários

- Descrição de uma utilização do sistema
 - Sequência de passos
 - Descrevem a interação entre o utilizador e o sistema

- Exemplo - Loja online
 - “The customer browses the catalog and adds desired items to the Shopping basket. When the customer wishes to pay, the customer describes the shipping and credit card information and confirms the sale. The System Checks the authorization on the credit card and confirms the sale both immediately and with a follow-up e-mail.”
 - Apenas um exemplo de uma situação que pode acontecer

 - Se o cartão não for autorizado
 - Outro cenário diferente

 - Se for um cliente habitual
 - Não é necessário pedir os dados de envio e pagamento
 - Outro cenário

Cenários

- Todos estes exemplos
 - São diferentes
 - Mas semelhantes
- Têm todos o mesmo objetivo
 - Comprar um produto
- Nem sempre a compra é bem sucedida
 - Mas o objetivo final mantém-se
- Use case
 - Conjunto de cenários
 - Relacionados entre si através de um objetivo comum

Engenharia de requisitos

- Deve responder às perguntas:
 - O sistema a desenvolver serve para quê?
 - Deve fazer o quê?
- Interessa definir que “funcionalidades” o sistema deve exibir
 - para quem...
- E não como se implementa

Use Cases

- Casos de uso
- Permitem captar e ilustrar os requisitos do sistema
- Representação do sistema
 - Comum a todos os stakeholders
- São cenários de utilização do sistema
 - Os cenários são instâncias dos Use Cases

Use Cases

- Não existem sistemas “isolados”
- Definem as fronteiras do sistema com o “exterior”
- Especificam interações com o “exterior”
 - Pode haver interação com humanos ou com outros sistemas
- Podem ser considerados diagramas de contexto

Elementos básicos

- Atores
 - Utilizadores do sistema
 - Utilizadores, clientes, gestores, etc...
 - Pode não ser humano
 - Quem (ou o que) interage com o sistema
- Use Cases (propriamente ditos)
 - Funcionalidades oferecidas aos atores
 - Atores executa Use Cases

Conteúdo de um Use Case

- Como escrever
 - Não existe um standard
- Deve incluir
 - Cenário principal de sucesso
 - Caso normal
 - Descritos através de uma sequência de passos numerados
- Outros cenários alternativos
 - Extensões
 - Descritos através de variações do cenário principal
 - Podem ser sucessos ou falhas

Conteúdo de um Use Case - Exemplo

- Name: Buy a Product
- Main Actor: Customer
- Main Success Scenario :
 1. Customer browses catalog and selects items to buy
 2. Customer goes to check out
 3. Customer fills in shipping information (address ; next-day or 3-day delivery)
 4. System presents full pricing information, including shipping
 5. Customer fills in credit card information
 6. System authorizes purchase
 7. System confirms sale immediately
 8. System sends confirmation email to customer
- Extensions :
 - 3a : Customer is regular customer
 1. System displays current shipping, pricing, and billing information
 2. Customer may accept or override these defaults, returns to MSS at step 6
 - 6a : System fails to authorize credit purchase
 1. Customer may re-enter credit card information or may cancel

Cada Use Case

- Tem um ator principal
 - Invoca o Use Case para ter acesso ao serviço
 - Ator que tem como objetivo
 - o que o Use Case satisfaz
 - normalmente, é quem inicia o Use Case
- Podem existir outros atores
 - Atores secundários

Use Case

Cada passo/etapa

- Representa uma interação entre o ator e o sistema
- Frases simples
- Descrever de forma clara quem executa o passo ou etapa
- Descrever de forma clara qual o objetivo do ator
 - Não interessa descrever os mecanismos para atingir o objetivo
 - Não interessa descrever o interface do utilizador

Use Case

- Cada extensão
 - Descreve uma alternativa ao caso normal
 - Descreve uma condição
 - Resulta numa interação diferente das descritas no cenário principal
 - Faz despoletar a alternativa
 - Descreve quais as diferenças para o cenário principal
- Como descrever uma extensão
 - Começar indicar qual o passo do cenário principal que pode dar origem à extensão
 - Incluir descrição da condição
 - Continuar com a lista de passos
 - Como no cenário principal
 - No final, indicar para onde se deve continuar
 - No cenário principal

Use Case

- Excelente ferramenta para fazer brainstorming
 - Sobre o cenário principal de sucesso
- Para cada passo/etapa perguntar:
 - O que pode correr de forma diferente?
 - O que pode correr mal?
- Perceber primeiro quais as condições das extensões
 - Só depois pensar nos detalhes de cada extensão
- Um passo/etapa complexa
 - Pode ser outro Use Case
 - Não existe forma standard para especificar ligações entre Use Cases
 - Exemplos: sublinhar o texto, hyperlinks, etc...

Use Case

- Pode incluir
 - Pré-condições: O que deve ser assumido como verdade, antes que o use case possa iniciado
 - Garantias: O que o sistema deve garantir depois do use case terminar
 - Garantias de sucesso: quando o use case termina de “normal”
 - Garantias mínimas: nos outros casos
 - Trigger: Que evento faz com que o use case seja iniciado

Como identificar atores?

- 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?

Identificar Use Cases

Guide-line

- Um Use Case deve corresponder a uma grande funcionalidade bem definida do sistema
 - Bem definida
 - Deve dar origem a “um resultado”
 - Não deve ser demasiado abstrata (ou de alto nível)
 - Não deve ser demasiado específica (ou de baixo nível)

Identificar Use Cases

Exemplo

- Numa ATM não interessa dizer que uma “grande funcionalidade bem definida” é:
 - “usar a ATM”!
- Mas também não interessa dizer que é:
 - “carregar na tecla dos 20€”
- Pode ser:
 - levantar dinheiro
 - pagar uma conta
 - ver o saldo
 - ativar a Via Verde
 - etc...

Diagramas de Use Cases

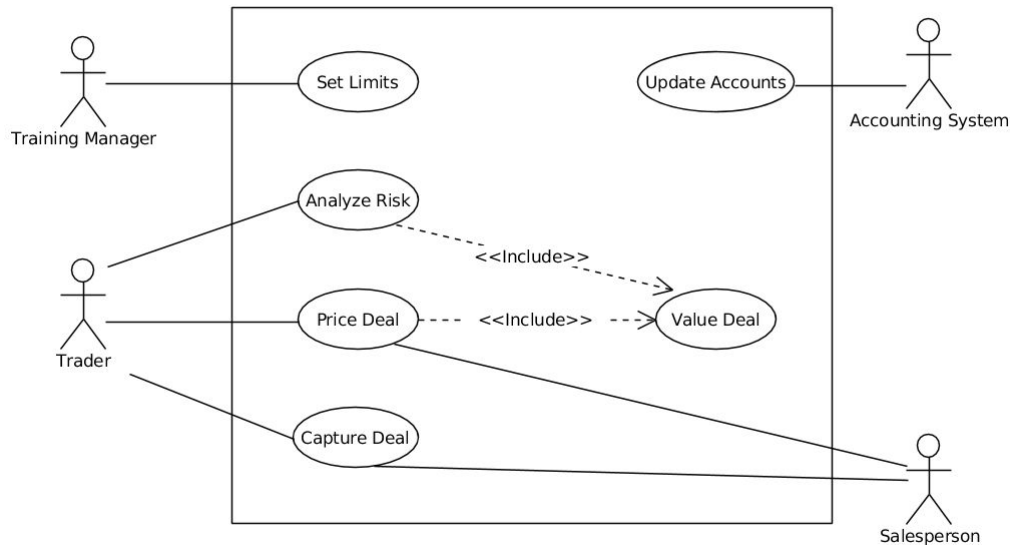
- UML
 - Não indica qual o conteúdo de um Use Case
 - Mas especifica um diagrama para apresentar os Use Cases
- Diagramas de Use Cases
 - São úteis
 - Mas não são obrigatórios
 - “Tabela gráfica” de conteúdos
 - Com todos os Use Cases do sistema
 - Diagrama de contexto
 - Mostra as fronteiras do sistema
 - E interações com o mundo “exterior”

Diagramas de Use Cases

- Mostram
 - Relações entre Atores e Use Cases
 - Que atores invocam quais Use Cases
 - Que Use Cases usam outros Use Cases
 - Relações entre Use Cases
 - Fronteira do sistema
- Relações entre Use Cases
 - *include* ou *uses*
 - *extends*
 - herança

Diagramas de Use Cases

Exemplo



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 (controle 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 pizzeria efetue reservas de mesa, verificando se este tem autorização para o efetuar. O mesmo deverá acontecer para os restantes empregados.

”

Atores

- Não fazem parte do sistema
- São representados no sistema
- São cruciais para a análise



Cliente



Empregado balcão



Gestor pizzeria

Use Cases

- Descrevem as grandes funcionalidades dos atores
- Devem “dar” um resultado observável e útil para um ator
- Um use case especifica o comportamento do sistema
 - ou parte do sistema
- Traduz-se num conjunto de sequências de ações
 - incluindo variantes

Efectuar
encomenda

Reservar
encomenda

Controlo de
acessos

Desconto
Internet

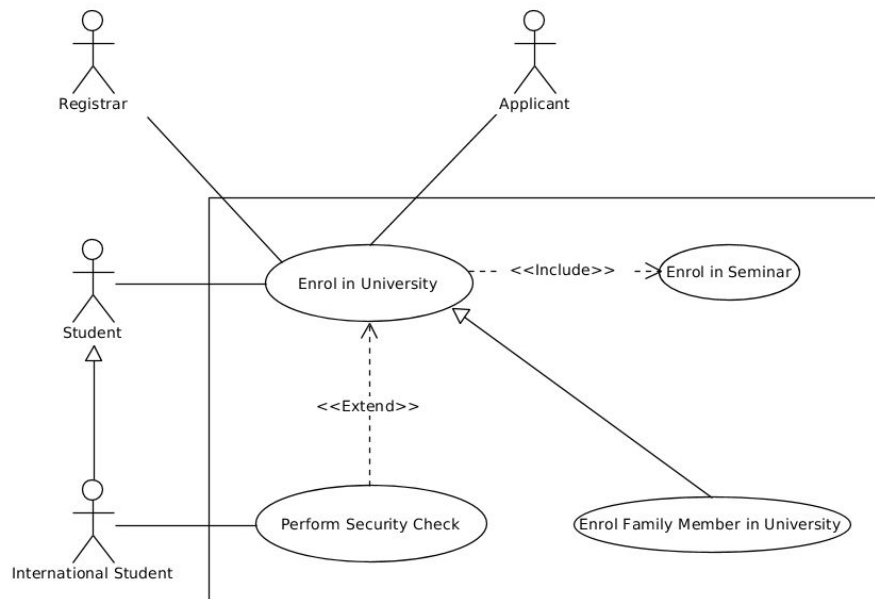
Associações

- Relações entre
 - Atores e atores
 - Atores e Use Cases
 - Use Cases e Use Cases
- Uma associação entre um Ator e um Use Case
 - Indica que há comunicação entre eles
 - Pode existir o envio/receção de mensagens

Associações

Tipos de associações

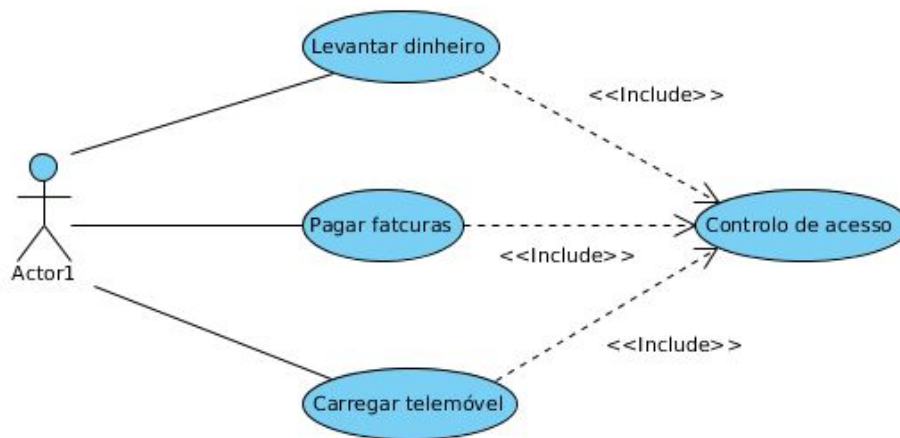
- “Normais”
 - Entre Atores e Use Cases
- Include ou Uses
 - Entre Use Cases
- Extend
 - Entre Use Cases
- Herança
 - Entre Atores e/ou Use Cases



Associações

<<include>> ou <<uses>>

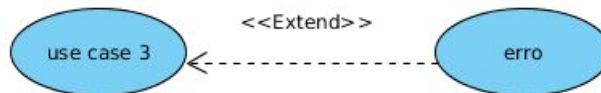
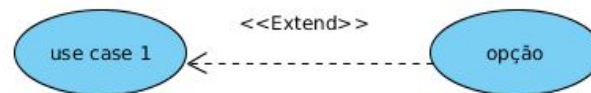
- Representar comportamento comum entre vários Use Cases



Associações

<<extend>>

- Representar
 - Variantes
 - Opções
 - Exceções
- Exemplo
 - Use case 1 tem uma opção que pode ser invocada
 - Use case 3 pode dar origem a um erro

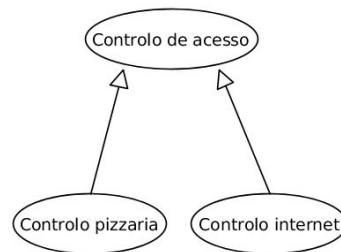
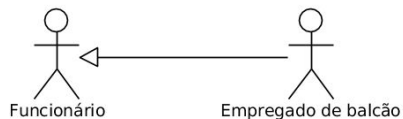


Associações: Uses e Extends

- “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.
 - Um **extends**... significa que o use case base incorpora implicitamente o comportamento 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.

Generalizações/Especializações

- Herança
- Aplicável a use cases e a atores
- São particularizações
 - têm um tronco comum: o use case ou ator geral;
- A generalização possui as mesmas propriedades duma relação pai/filho
 - use case “filho” herda ou substitui por completo o comportamento do use case “pai”



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 (controle 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 pizzeria efetue reservas de mesa, verificando se este tem autorização para o efetuar. 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 (**controle 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 pizzeria efetue reservas de mesa, verificando **se este tem autorização** para o efetuar. **O mesmo** deverá acontecer para os restantes **empregados**.

”

Ator	Use Case
Cliente	Efectuar encomenda Internet Desconto Controlo de acesso
Gestor pizzeria	Efectuar encomenda Controlo acesso
Gestor pizzeria	Reservar mesa Controlo acesso

Exemplo - Encomendar uma pizza

- Nome: Encomendar uma pizza
- Ator principal: Cliente
- Comportamento normal :
 1. O cliente valida-se perante o sistema;
 2. O Use Case começa quando o cliente seleciona 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 - Encomendar uma pizza

- O que é que pode variar?
 - Código de produto inexistente.
 - Cartão de crédito não válido.
 - Se o produto está em promoção, calcular desconto
- Extensões
 - 4. O cliente adiciona produtos à encomenda através do código
 - 4a. se o código não existe o sistema avisa e volta ao passo 3.
 - 5. “O sistema ‘completa’ com a descrição e o preço do produto”
 - 5a. “Se o produto está em promoção, calcular desconto”
 - 1. O sistema retorna o valor do desconto que mostra ao cliente
 - 2. Calcula o desconto subtraindo ao preço do produto e volta ao passo 6
 - 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

Exemplo - Encomendar uma pizza

Ações comuns

- Considerando o passo:
 1. “O cliente valida-se perante o sistema”
- “Validação perante o sistema” pode ser comum a vários Use Cases;
 - É um procedimento reutilizável noutros casos ou que já existia no sistema
 - Relação de <<uses>> ou <<include>>
 - Com o Use Case “Validação perante o sistema”

Outras utilizações

- Ajudam a conceber e validar a arquitetura do sistema
 - a estrutura dos use cases é determinante
- Podem ser uma base para os testes do sistema;
- São um “ponto de referência” em todas as etapas de desenvolvimento

Importante

- Os Use Cases descrevem um sistema a um nível (muito) elevado de abstração
- Não se diz como serão implementados
- É uma disciplina de análise difícil
- É fácil resvalar para a tecnologia
- É fácil descer de nível de abstração
- Cuidado com as especializações, *extends* e *include/uses*

Bibliografia

- UML Distilled A Brief Guide to the Standard Object Modeling Language. Martin Fowler. 3rd edition. Addison-Wesley Professional. 2003. Capítulo 9.
- Software Engineering. Ian Sommerville. 10th Edition. Addison-Wesley. 2016. Capítulo 5.