

# Casos de Uso e Respectivos Diagramas

João Paulo Barros

Escola Superior de Tecnologia e Gestão  
Instituto Politécnico de Beja

22 de Maio de 2009

Este texto apresenta de forma resumida os **CASOS DE USO** [1, 3], mais especificamente o que são e como devem ser especificados, quer textualmente, o mais importante, quer sob a forma de um tipo diagramas UML apropriadamente denominados **DIAGRAMAS DE CASOS DE USO** [2].

*casos de uso*

*diagramas de casos de uso*

## 1 Casos de Uso

O objectivo dos casos de uso é a identificação das funcionalidades requeridas para o sistema. Assim, os casos de uso incluem-se na fase de **ANÁLISE DE REQUISITOS**, a fase em que procuramos identificar, da melhor forma possível, o que é que o nosso sistema (frequentemente um ou mais programas) deve realmente ser capaz de fazer. Como seria de esperar, esta fase do desenvolvimento do software tem a sua própria terminologia e é por aí mesmo que vamos começar, apontado nomes e o que eles significam. Aqui ficam então os que nos interessam:

*análise de requisitos*

**Sistema em discussão (*system under discussion*)** O sistema que se pretende entender, modelar ou desenvolver. Pode ser um pequeno programa ou um sistema constituído por que vários programas e várias máquinas e tipos de utilizador.

**Acção (*action*)** Algo que é executado (feito) por alguém ou por algum sistema informático. Uma unidade de comportamento.

**Cenário (*Scenario*)** Sequência de acções expressa sem *ifs* ou caminhos alternativos.

**Actor (*actor*)** Alguém ou algo que exhibe um comportamento, ou seja, que executa acções. Corresponde sempre a uma entidade que está fora do sistema em discussão.

**Caso de uso (*use case*)** Identificação e especificação (captura) de um conjunto de cenários de utilização do sistema em discussão.

**Parte interessada (*stakeholder*)** Alguém com interesse no sistema em discussão. Por exemplo, as pessoas que o utilizam, o dono do mesmo, pessoas que utilizem os resultados do sistema, outros programas.

Cada caso de uso deve ser especificado utilizando texto. Para que todos os casos de uso contenham o mesmo tipo de informação, é importante utilizar uma minuta (*template*) para cada um. Aqui fica uma em que se assinalam com asterisco as partes de preenchimento obrigatório:

- \*Nome** Um nome que transmita para que serve este caso de uso. Deve começar por um verbo.
- Breve descrição** Uma descrição muito resumida que indique os objectivos do caso de uso.
- \*Cenário principal** Onde se especifica a sequência de acções.
- \*Cenários adicionais** Onde se especificam os vários cenários adicionais.
- Actor Principal** Indicação do actor principal para este caso de uso.
- Partes interessadas e respectivos interesses** A quem interessa este caso e uso e porque é que interessa.
- Pre-condições (*pre-conditions*)** O que tem de ser verdade para este caso de uso poder ser realizado.
- Pós-condições (*pre-conditions*)** O que tem de ser verdade para este caso de uso depois de estar concluído.
- Condição de sucesso** O que tem de ser verdade para considerarmos o caso de uso bem sucedido
- Condição mínima de sucesso** O que tem de ser verdade para considerarmos o caso de uso minimamente bem sucedido
- Dependências** Outros casos de uso de que este depende.

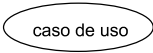

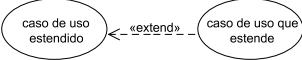
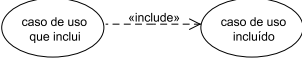
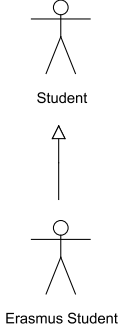
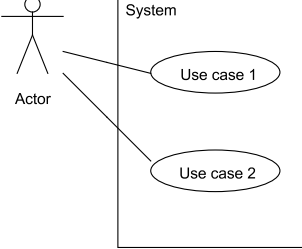
Na especificação de cada caso de uso o objectivo é o de especificar tudo o que o actor pode fazer. Frequentemente, este será um utilizador, mas pode ser um outro programa. Assim, para cada caso de uso o mais importante é atingir este objectivo. Tal pode ser feito seguindo a seguinte receita, baseada na apresentada em [3]:

1. Pergunte *o que acontece?* Tal fornecerá o início do cenário principal.
2. Pergunte *e depois o que acontece?* e continue a perguntar até completar o cenário principal. Não coloque as alternativas. Essas ficam para o passo seguinte.
3. Pergunte *o que mais pode acontecer?* Tal fornecerá os cenários alternativos. Continue a perguntar até ter todos os cenários alternativos. É muito importante que identifique todos os cenários alternativos e os complete o mais possível. Descobri-los mais tarde implicará muito mais trabalho.

Conforme já referido, a especificação dos casos de uso utilizando texto, ou seja o preenchimento das respectivas minutas, é o fundamental. No entanto, a representação gráfica tem a vantagem de permitir uma visão geral dos vários actores e casos de uso, bem como das principais relações entre eles. A secção seguinte resume o que mais interessa sobre os diagramas de caso de uso da linguagem UML.

## 2 Casos de Uso e UML

A seguinte tabela identifica as várias notações dos diagramas de casos de uso da linguagem UML que nos vão interessar.

Notação gráfica	Significado
	Um caso de uso
	Um actor (pessoa, programa ou peça de hardware). Também pode ser representado por uma pequena imagem. Um actor representa um "papel" na sua relação com o sistema. Este papel pode ser desempenhado por uma pessoa, programa ou peça de hardware.
	O comportamento do caso de uso estendido <b>pode ser</b> estendido pelo comportamento de outro. O caso de uso estendido é definido independentemente do caso de uso que o estende, e faz sentido mesmo sem este.
	O caso de uso incluído faz parte do caso de uso que o inclui. O caso de uso incluído não é opcional sendo sempre necessário para a correcta execução do caso de uso que o inclui. Esta relação deve ser utilizada quando existem partes comuns em mais do que um caso de uso. Esta parte comum deve passar a constituir um novo caso de uso que é incluído por todos os casos de uso que têm essa parte em comum. O caso de uso incluído pode ser visto como uma função que é chamada/utilizada pelo caso de uso que o inclui. O sentido da seta transmite esta ideia em que o que inclui depende do incluído, mas este não depende daquele que o inclui.
	Um actor (no exemplo, Erasmus Student) que é um caso particular de outro (Student). Note que o Erasmus Student é um Student pelo que faz tudo o que o Student faz.
	Pode utilizar-se um rectângulo que especifica o sistema ou sub-sistema a que os casos de uso pertencem.

### 3 Ferramentas

Existem MUITAS ferramentas para a linguagem UML. A wikipedia apresenta um boa lista em [http://en.wikipedia.org/wiki/List\\_of\\_UML\\_tools](http://en.wikipedia.org/wiki/List_of_UML_tools).

Se se pretender apenas desenhar, a UMLet (<http://www.umlet.com/>) é especialmente simples e eficiente.

### Referências

- [1] Alistair Cockburn. *Writing Effective Use Cases*. Addison-Wesley, 2001. 1
- [2] OMG. *OMG Unified Modeling Language™ (OMG UML), Superstructure — Version 2.2*. OMG Document Number: formal/2009-02-02, 2009-02-02. 1
- [3] Doug Rosenberg and Kendall Scott. *Applying Use Case Driven Object Modeling with UML: An Annotated e-Commerce Example*. Addison-Wesley, 2001. 1, 2