



# Desenvolvimento de Sistemas Software

Aula Teórica 5

## Diagramas de Use Case



# Use Cases

*“To my knowledge, no other software engineering language construct as significant as use cases has been adopted so quickly and so widely among practitioners. I believe this is because use cases play a role in so many different aspects of software engineering”*

*Use Cases - Yesterday, Today, and Tomorrow*

Ivar Jacobson (*‘pai’ dos Use Cases*)

(e também, Vice President

Process Strategy

Rational Software

IBM Software Group)



# Exemplo - Máquina Multibanco

- **Cenários**

...

- **Use Cases**

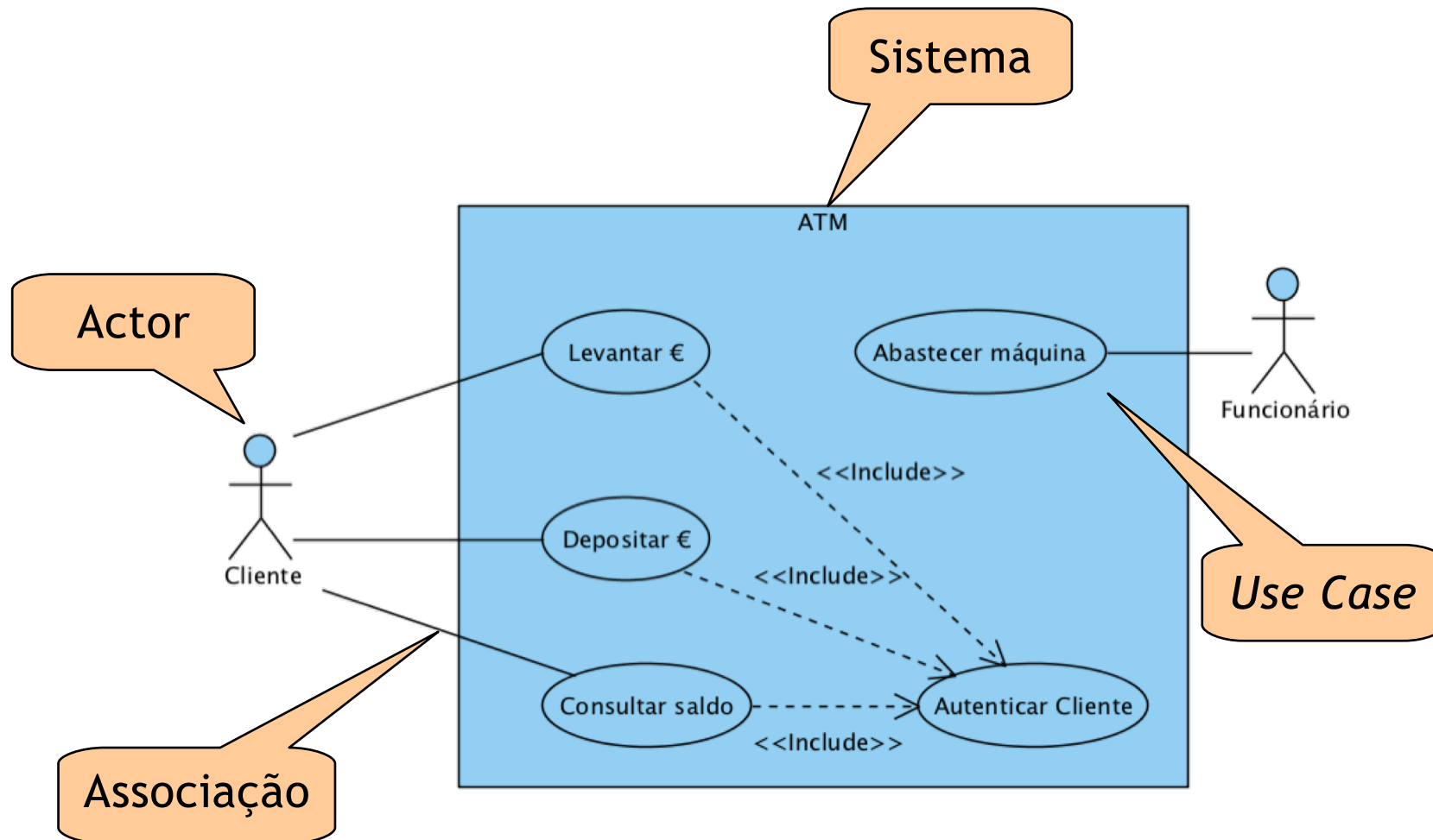
- Levantar €
- Pagar serviço
- Efectuar transferência
- Carregar máquina

- **Actores**

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



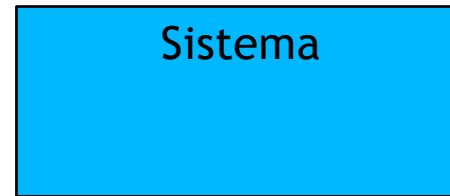
# Diagrama de Use Case - Exemplo





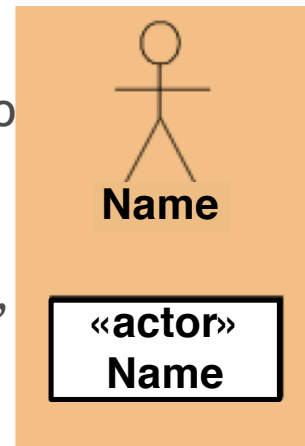
## Sistema

- define as fronteiras da solução a desenvolver



## Actor

- uma abstracção para uma entidade fora do sistema
- um actor modela um propósito (alguém que tem um interesse específico no sistema) - pode não mapear 1 para 1 com entidades no mundo real
- um actor não é necessariamente um humano - pode ser um computador, outro sistema, etc.
- cada actor representa um papel (“role”) que “alguém” ou qualquer “coisa” externa ao sistema pode assumir
- o conjunto de todos os actores definem todas as formas de interacção com o sistema



## Associação

- representa comunicação entre o actor e o sistema - através de *use cases*
- pode ser bi-direccional ou uni-direccional



# Identificação de Use Cases

Etapas a cumprir (com o auxílio de cenários de utilização do sistema):

1. Identificar actores (quem utiliza o sistema)
2. Identificar *use cases* (o que se pode fazer no sistema)
3. Identificar associações (quem pode fazer o quê)

## *Identificar actores*

- Quem vai utilizar o sistema?
- Neste caso: Cliente, Bancário, Técnico de Manutenção?, Impressora?, Servidor do banco?

## *Identificar Use Cases*

- Objectivos dos utilizadores/actores?
- Resposta a estímulos externos.



# Que Actores? Que Associações?

## *Identificar associações*

- Que actores utilizam que *Use Cases*?
- Nem sempre é imediatamente evidente se a comunicação entre o sistema em análise e sistemas externos deve ser representada. Quatro abordagens podem ser identificadas:
  - X mostrar todas as associações;
  - X mostrar apenas as associações relativas a interacção iniciada por sistemas externos;  
mostrar apenas as associações relativas a interacções em que é o sistema externo o interessado no *use case*;
  - X não mostrar associações com sistemas externos.



# Que Actores? Que Associações?

## *Todas as associações*

- Todos os sistemas externos que interagem com o sistema em análise são apresentados como actores e todas as interacções são representadas nos diagramas.
- Demasiado abrangente, em muitos casos existem interacções com outros sistemas apenas por razões de implementação e não por se tratarem de requisitos do sistema.

## *Apenas as associações relativas a interacção iniciada por sistemas externos*

- Só são representados como actores os sistemas externos que iniciem diálogo com o sistema em análise.
- Mesmo assim muito abrangente.





# Que Actores? Que Associações?

## *Apenas as associações em que é o sistema externo o interessado*

- Neste caso só são apresentados como actores os sistemas externos que necessitam de funcionalidade fornecida pelo sistema em análise.
- Usalmente esta é uma solução equilibrada.

## *Não mostrar associações com sistemas externos*

- Apenas os utilizadores são actores, neste caso quando existem sistemas externos apresentam-se os seus actores em diálogo directo com o sistema a ser modelado.
- De uma outra forma esta solução também é demasiado abrangente e pode levar a confusão sobre quem está realmente a utilizar o sistema.



## Use Cases - resumo até agora

- Forma sistemática de capturar requisitos funcionais
  - que serviços deve fornecer; a quem os deve fornecer
  - mas **não** suportam a captura de requisitos não funcionais
- Notação diagramática facilita o diálogo
  - com os clientes e dentro da equipa de desenvolvimento
- Modelam o contexto geral do sistema
  - Quais os actores que com ele se relacionam
- Especificam todas as possíveis utilizações
  - O que cada actor pode fazer no sistema
- O centro de todo o processo de desenvolvimento
  - Desde a concepção da arquitectura, passando pelos testes, até ao manual de utilização...



## Use Cases - resumo até agora

- A concepção do sistema é guiada pelo modelo de *Use Case*:
  - Utilizando diagramas de *Use Case*, clientes e equipa de desenvolvimento podem chegar a um acordo sobre qual o sistema a desenvolver
- A implementação do sistema é guiada pelo modelo de *Use Case*:
  - cada *Use Case* é implementado sucessivamente:
  - quando todos os *Use Cases* estiverem implementados obtém-se o sistema final;
  - fica facilitada a manutenção sempre que os requisitos sejam alterados;
- O modelo de *Use Case* é utilizado para o planeamento de testes:
  - Após a definição do modelo de *Use Case*: planejar *black-box testing*.
  - Após a implementação dos *Use Cases*: planejar *white-box testing*.



### *Black-box testing*

- Utilizado para verificar se o sistema implementa toda a funcionalidade pretendida.
- Permite detectar erros de “omissão” (funcionalidade não implementada).

### *White-box testing*

- Utilizado para verificar se o sistema implementa a funcionalidade de forma correcta.
- Permite detectar erros na implementação da funcionalidade pretendida.



# Diagramas de Use Case

## Sumário:

- Notação base dos Diagramas de Use Case
  - Sistema
  - Use Cases
  - Actores
  - Associações
- Identificação de actores, use cases e associações