

# ENGENHARIA DE SOFTWARE APLICADA

## Use Cases

Licenciatura em Engenharia Informática

ESTSetúbal - IPS 2020/21

**Nuno Pina Gonçalves**

**Paulo Fournier**

**Rui Borges**

[nuno.pina@estsetubal.ips.pt](mailto:nuno.pina@estsetubal.ips.pt)

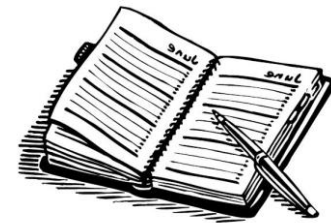
[paulo.fournier@estsetubal.ips.pt](mailto:paulo.fournier@estsetubal.ips.pt)

[rui.borges@estsetubal.ips.pt](mailto:rui.borges@estsetubal.ips.pt)

# Agenda

2

- O que é modelação com **use cases** (casos de uso)
- Diagrama de **use cases**
  - ▣ Elementos e notações gráficas
  - ▣ Relacionamentos
  - ▣ Como identificar os elementos durante o levantamento de requisitos
- Exercícios.



# Modelação com Use Cases

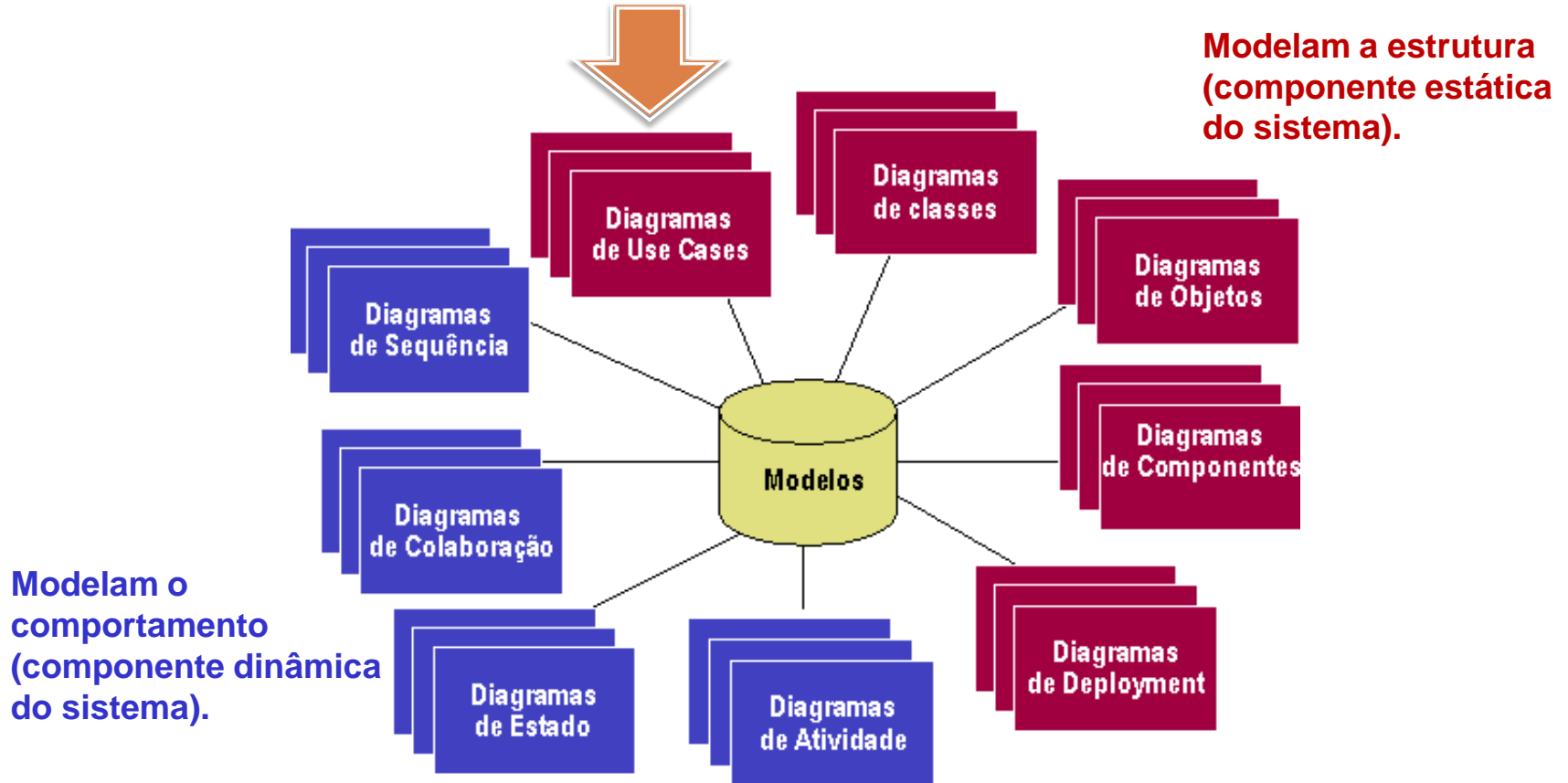
3

- A modelação com **use cases** é uma abordagem utilizada na engenharia de requisitos
- Suporta o processo de **levantamento dos requisitos funcionais** do sistema
- Cria-se um modelo onde são identificadas **as funcionalidades que o sistema deverá apresentar para cada um dos seus utilizadores.**
- Utiliza-se o **diagrama de use cases** da **Unified Modelling Language (UML)** para representar o modelo.

# UML - Diagramas

4

Suportam o processo de análise e especificação de Requisitos



# Modelação com Use Cases

5

- A modelação com **use cases** é muito útil quando:
  - ▣ O sistema é dominado por requisitos funcionais
  - ▣ O sistema tem diversos tipos de utilizadores a quem fornece diferentes funcionalidades
  - ▣ O sistema tem muitos interfaces
- A modelação com **use cases** não é adequada para capturar restrições do sistema (requisitos não funcionais).

# Modelação com Use Cases

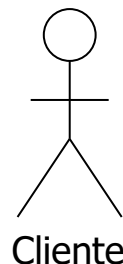
6

- Elementos a incluir no modelo
  - ▣ O quê ou quem utiliza o sistema.
  - ▣ Quais as funções que o sistema deverá oferecer aos seus utilizadores.
  - ▣ Onde se situa a fronteira do sistema.

# Diagrama de Use Case

7

- Ator (O quê ou quem)
  - ▣ Um ator interage diretamente com o sistema
  - ▣ Os atores identificam quem utiliza o sistema
  - ▣ Um ator especifica um papel ("role") desempenhado por uma entidade externa quando interage com o sistema
  - ▣ Um ator pode ser outro sistema



# Diagrama de Use Case

8

## □ **Use Cases** (funções do sistema)

- ▣ Um *Use Case* representa uma funcionalidade do sistema.
- ▣ Um **Use Case** é sempre iniciado por um ator - ator principal
- ▣ Para que o sistema consiga realizar a funcionalidade poderá ser necessário envolver outros atores - atores secundários
- ▣ Os identificadores são sempre escritos do ponto de vista dos atores



Efetuar Encomenda

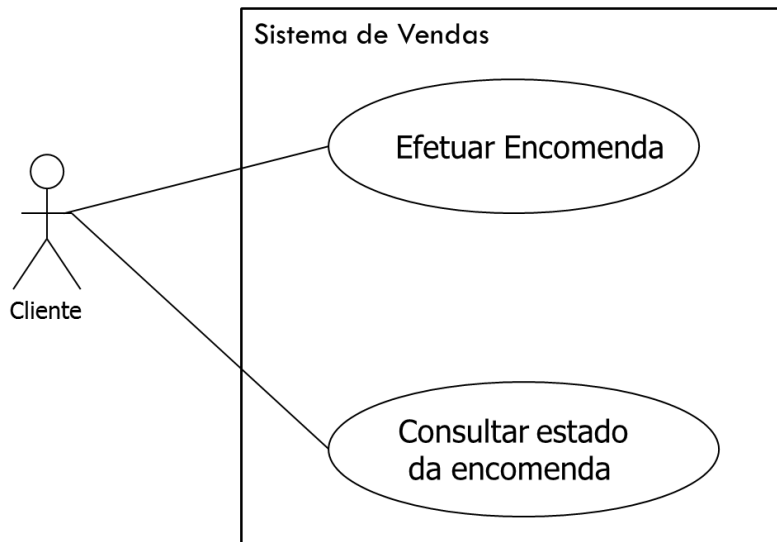
Consultar estado  
da encomenda



# Diagrama de Use Case

9

- Fronteira (onde se situa a fronteira)
  - ▣ Identifica-se o sistema e a sua fronteira.
  - ▣ No interior colocam-se as funcionalidades do sistema
  - ▣ No exterior colocam-se os utilizadores do sistema



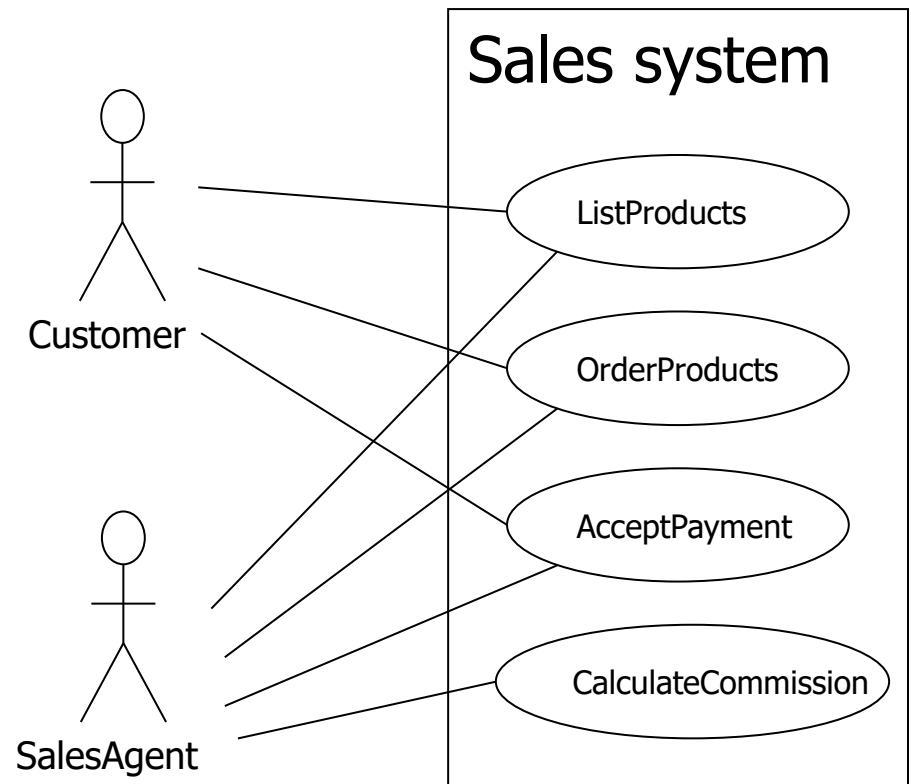
subject

SystemName

# Diagrama de Use Case

10

- Os atores Customer e SalesAgent são muito semelhantes
- Ambos interagem com ListProducts, OrderProducts e AcceptPayment
- O SalesAgent interage ainda com CalculateCommission
- O nosso diagrama está uma confusão – Conseguiremos simplificá-lo?

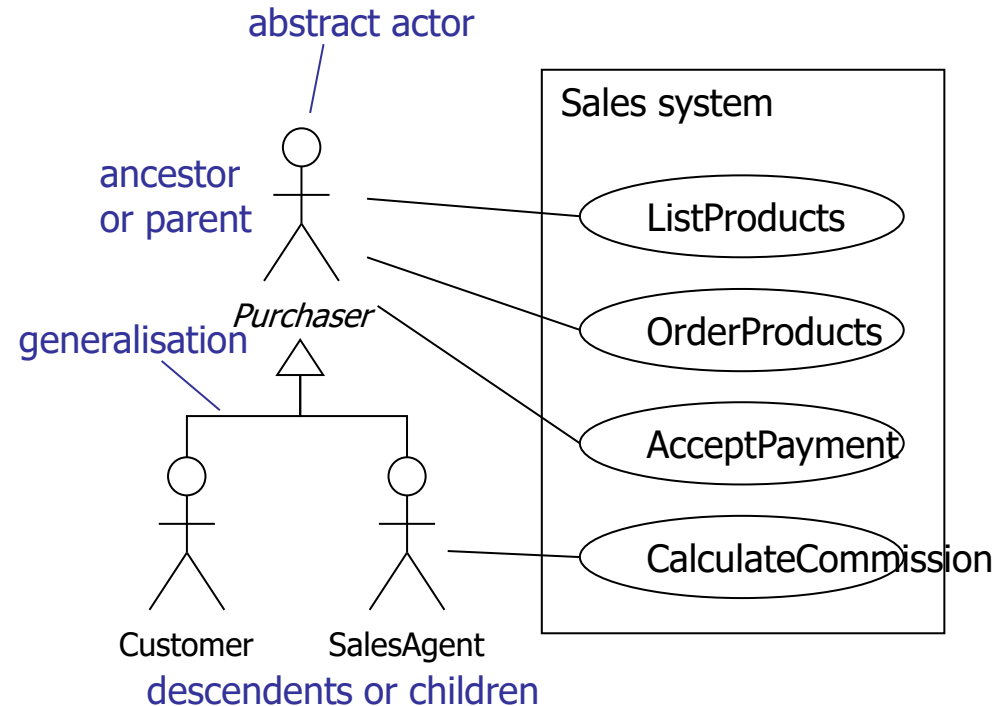


# Diagrama de Use Case

11

## □ Generalização de atores

- ▣ Se dois atores comunicarem da mesma forma com o mesmo conjunto de **use cases**



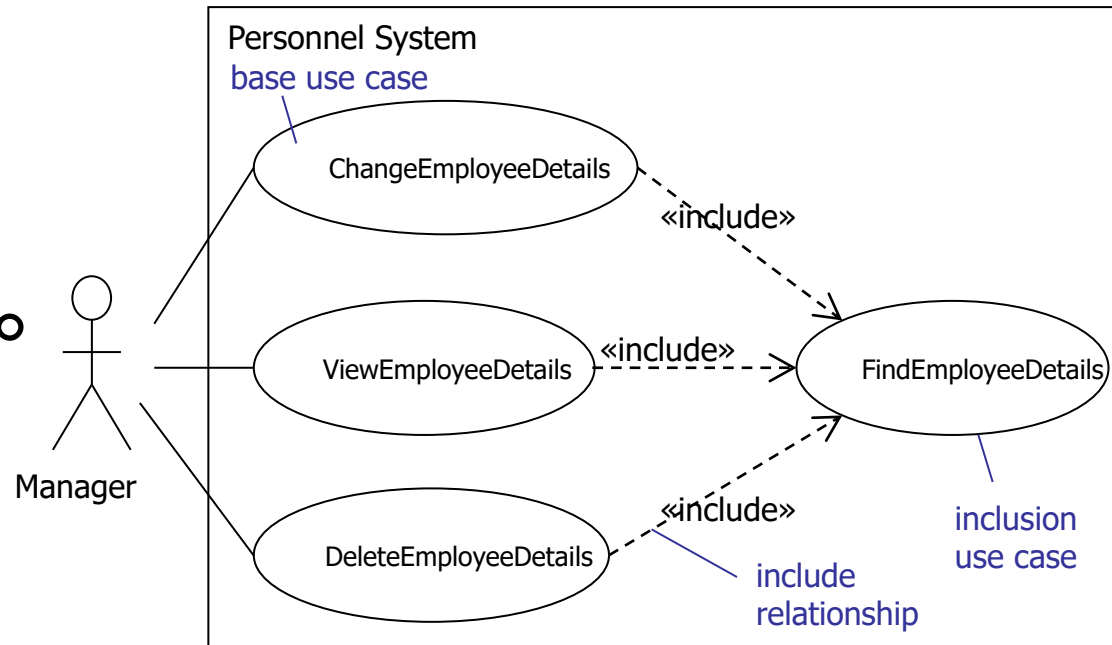
Utilizar a generalização de atores quando simplificar o modelo

# Diagrama de Use Case

12

## Relacionamento de <<include>>

- O use case incluído terá sempre que ser realizado



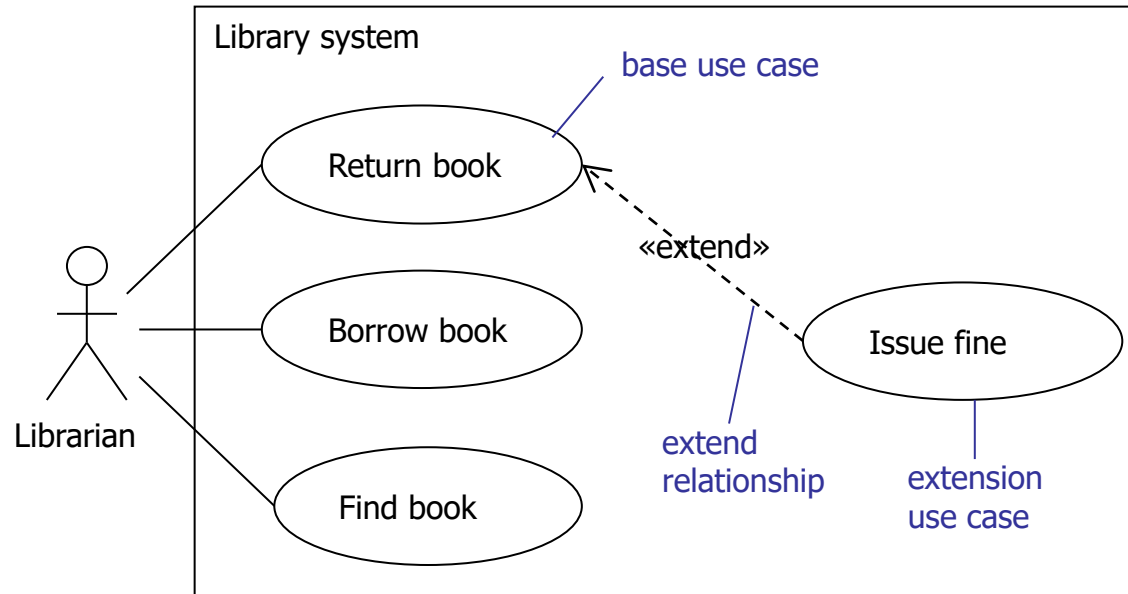
Quando os use cases partilham um comportamento comum podemos separar esse comportamento e incluí-lo nos use cases base.

# Diagrama de Use Case

13

## □ Relacionamento de <<extend>>

- O **use case** base, poderá despoletar a execução do use case de extensão
- Existe um comportamento que apenas será realizado quando estiverem reunidas determinadas condições



- "extension use case" insere comportamento no use case base.
- **use case** base fornece pontos de extensão.

# Minutos do desafio

14

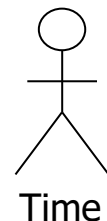
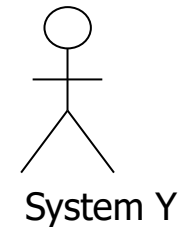
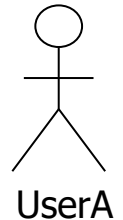


40 minutos...

# Identificação dos atores

15

- Algumas questões a dar resposta durante o levantamento de requisitos:
  - ▣ Quem utiliza o sistema?
  - ▣ O sistema terá que interagir com outros sistemas? Quais?
  - ▣ Quais os papéis que eles desempenham na interação?
  - ▣ Quem instala o sistema?
  - ▣ Quem inicia e encerra (shutdown) o sistema?
  - ▣ Quem faz a manutenção do sistema?
  - ▣ Quem recebe e/ou fornece informação ao sistema?
  - ▣ Existe alguma coisa que aconteça num determinado instante de tempo?



# Identificação dos Use Cases

16

- Inicia-se com a lista dos atores que interagem com o sistema.
- Quando se identifica **Use Cases**, pergunta-se
  - ▣ Quais as funções que um determinado ator irá necessitar do sistema?
  - ▣ O sistema armazena ou fornece informação? Se sim, quais os atores que despoletam/iniciam o comportamento ?
  - ▣ Existem eventos externos que afectem o sistema? Como é que o sistema toma conhecimento da ocorrência desses eventos?



# A reter

17

- A modelação com **use cases** é muito útil quando:
  - ▣ O sistema é dominado por requisitos funcionais
  - ▣ O sistema tem diversos tipos de utilizadores a quem fornece diferentes funcionalidades
  - ▣ O sistema tem muitas interfaces
- A modelação com **use cases** não é adequada para capturar restrições do sistema (requisitos não funcionais).
- Elementos a incluir no modelo
  - ▣ Os atores - O quê ou quem utiliza o sistema
  - ▣ Os use cases - Quais as funções que o sistema deverá oferecer aos seus utilizadores
  - ▣ Os relacionamentos entre os atores e os use cases
  - ▣ A fronteira do sistema