

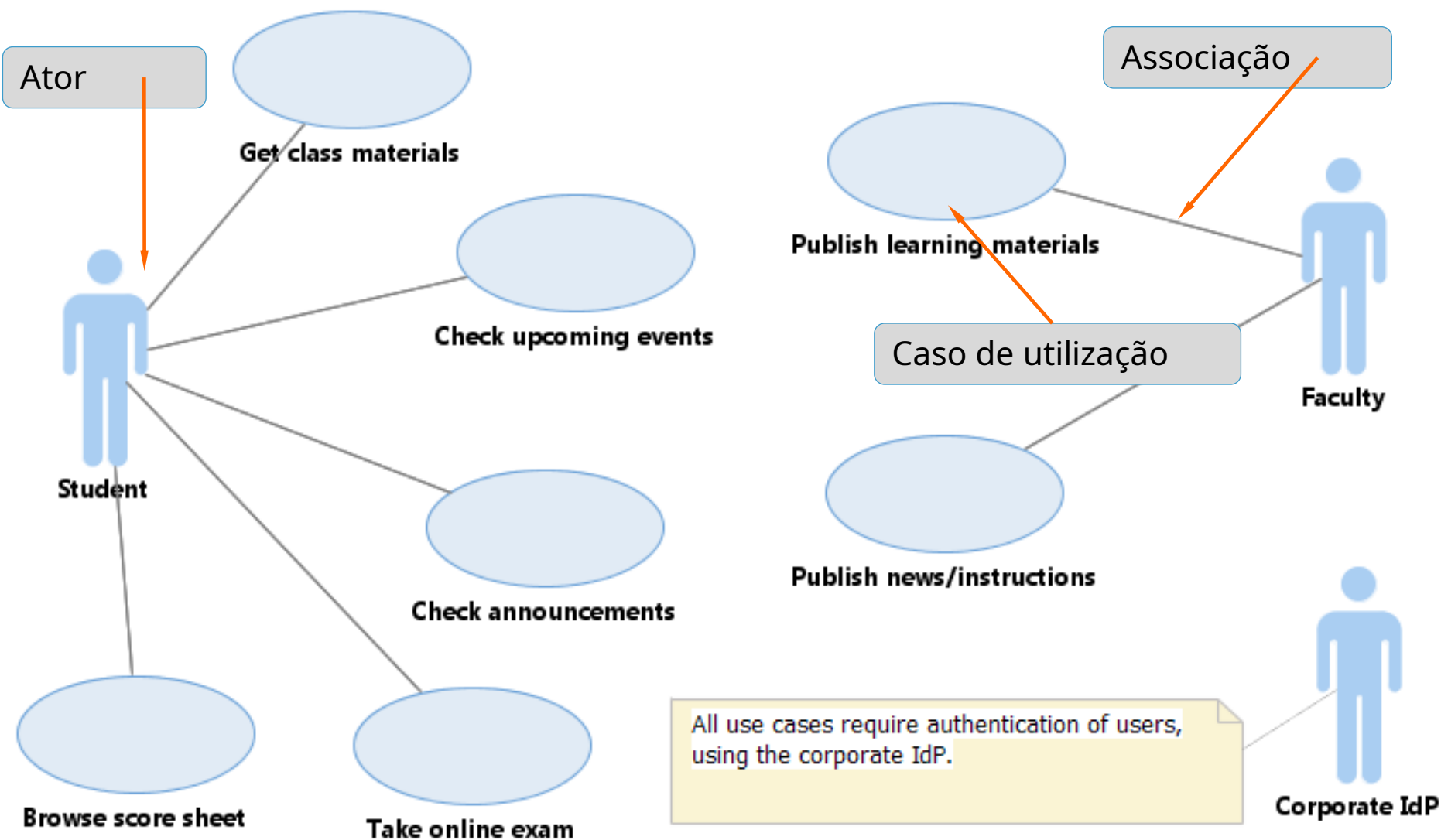
40431: Modelação e Análise de Sistemas

# Modelação com casos de utilização: complemento

Ilídio Oliveira

v2022-10-07

# Elementos do Diagrama de Casos de Utilização



# Reutilização de comportamentos com include

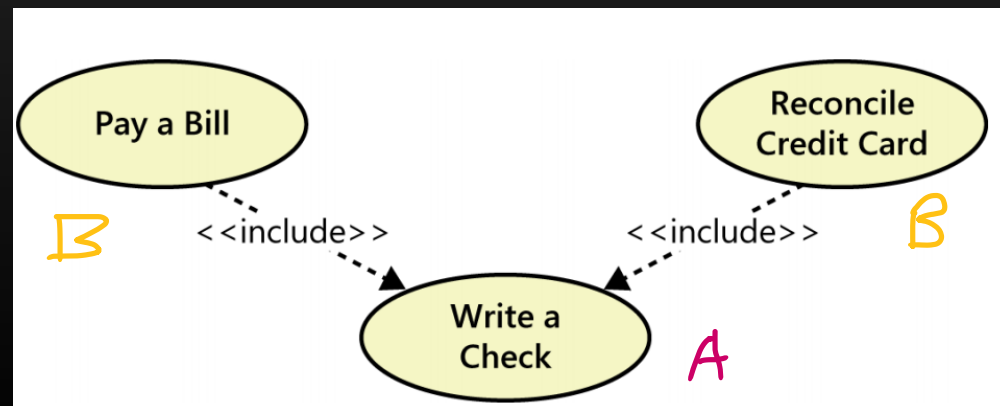
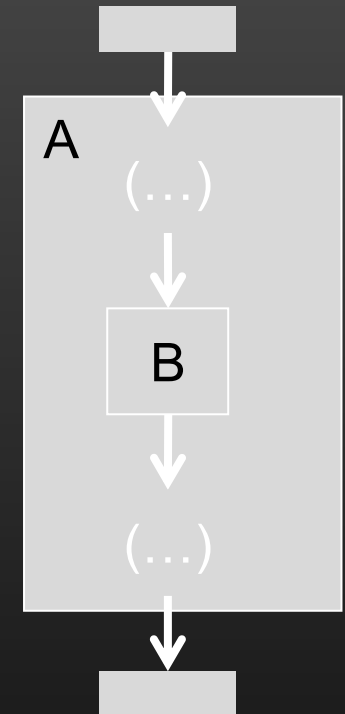
## A história A inclui a história B

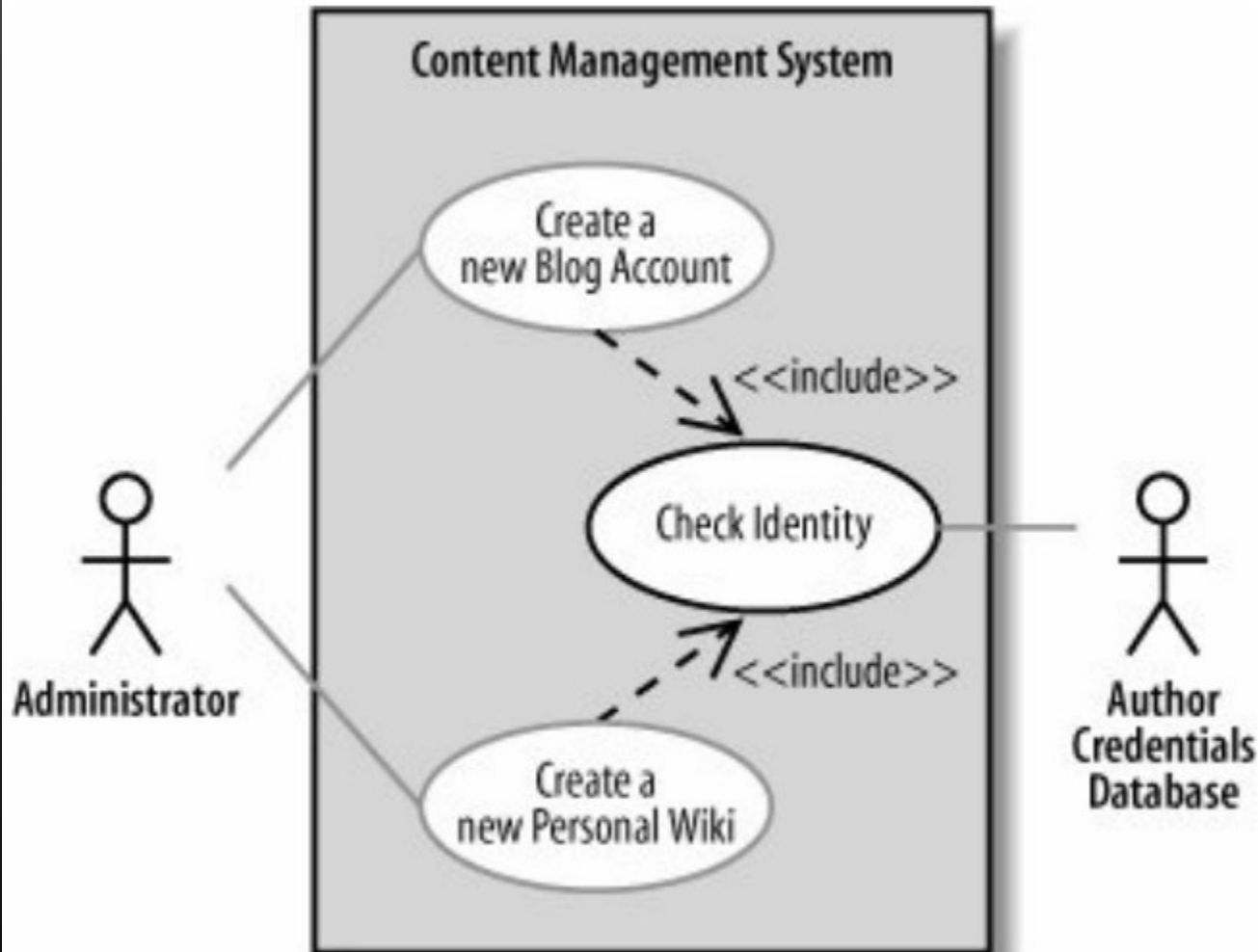
O comportamento em A inclui sempre o comportamento modelado em B

Pode ajudar a “colocar e, evidência” do comportamento comum

Include é o estereótipo da relação de dependência

- Não é um verbo conjugado



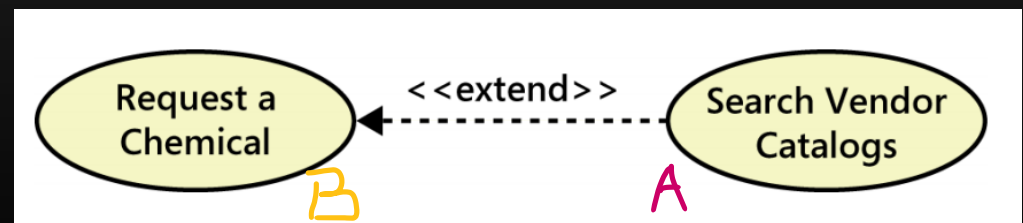
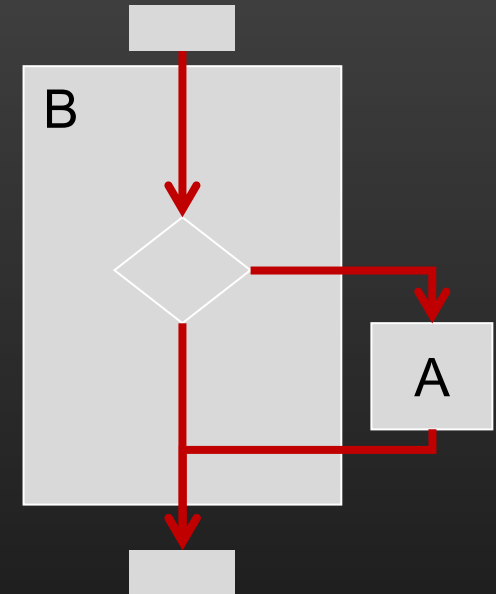


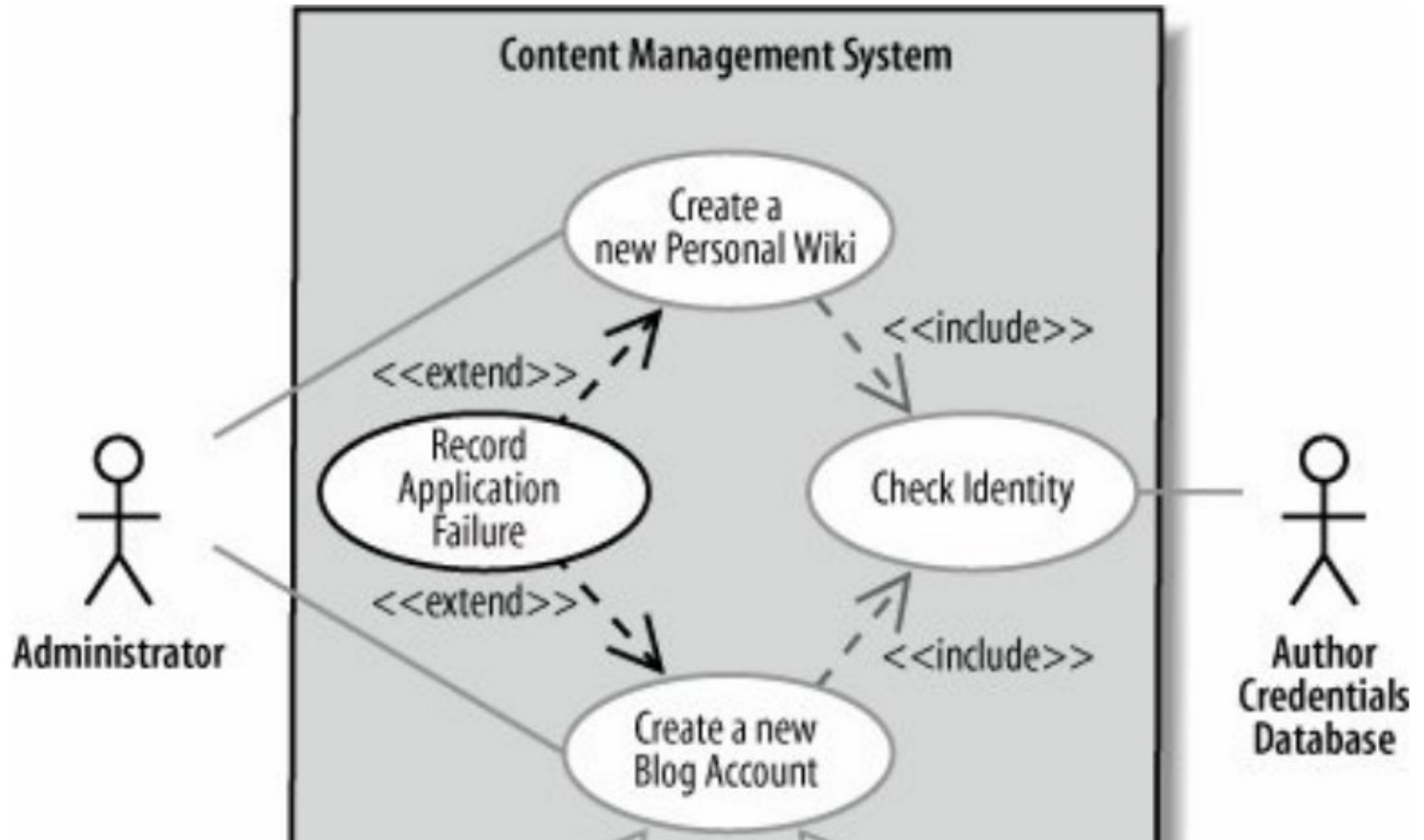
# Activação de comportamento opcional com extend

## A história A pode ampliar a história B

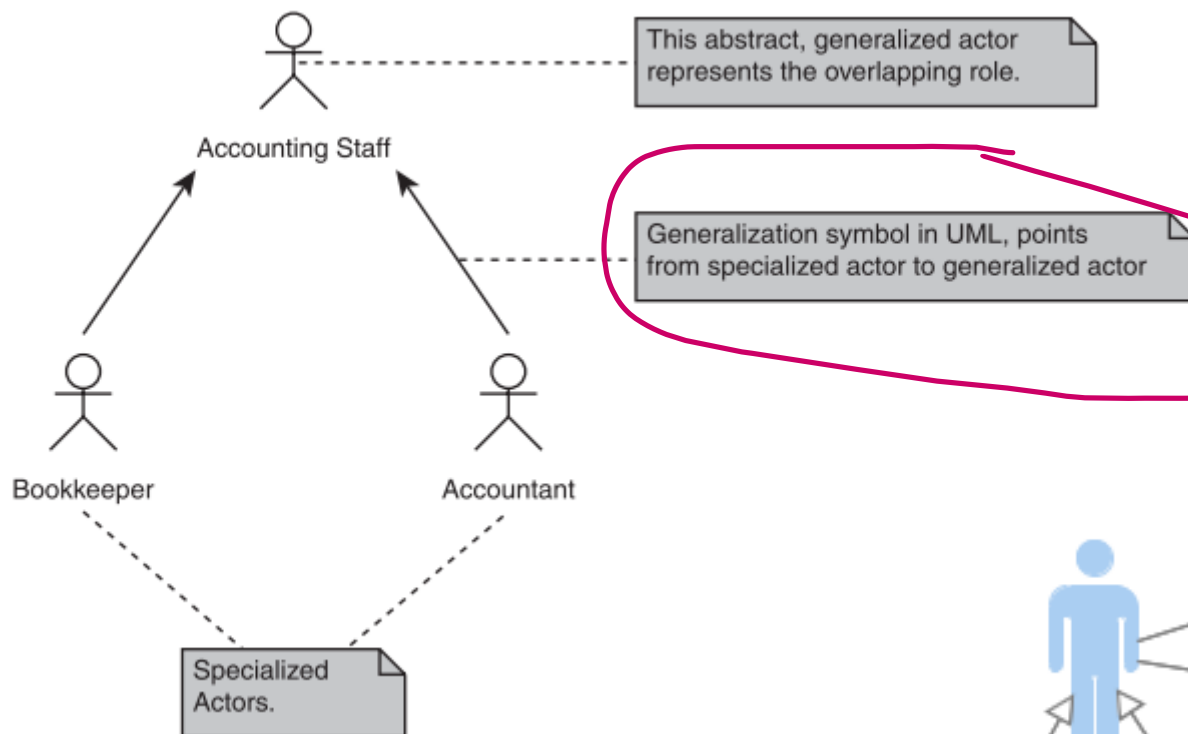
O comportamento de B pode incorporar o comportamento de A, dependendo da verificação de uma "condição de extensão" (ponto de extensão)

Ao contrário da relação include, a relação extend modela comportamento opcional / condicional

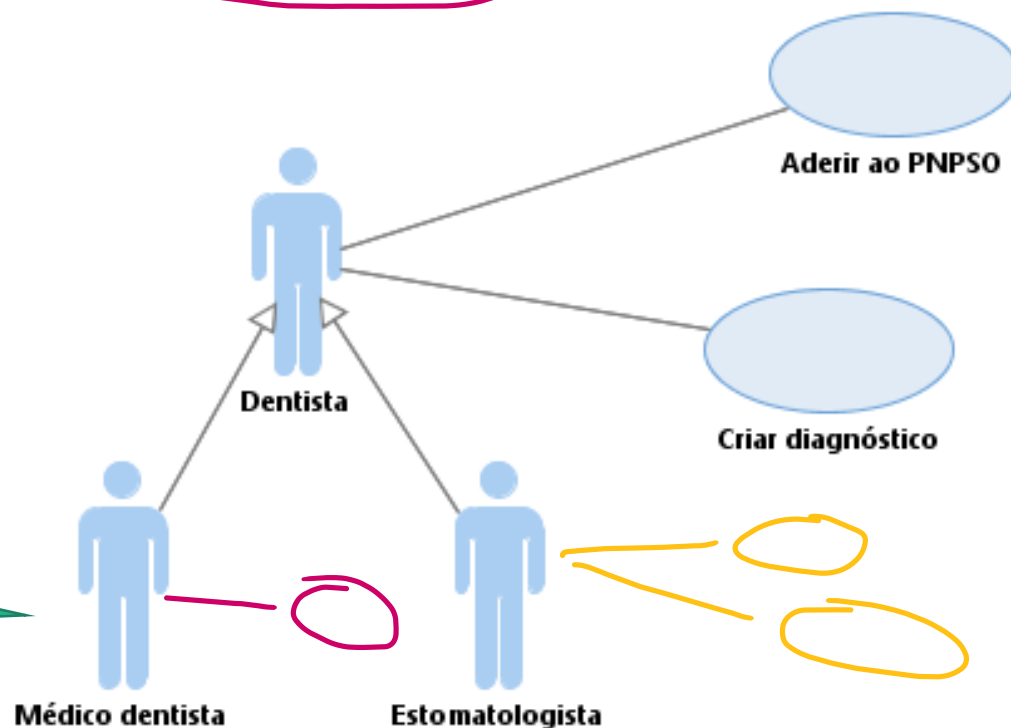




# Pode haver generalização entre atores (i.e., papéis)



O actor "especializado" assume todas as caraterísticas / associações do actor "geral", ao qual acrescenta especificidades



# Situações de modelação com CaU

universidade de aveiro  
departamento de eletrónica,  
telecomunicações e informática



deti



# Atores primários e secundários

## Ator primário

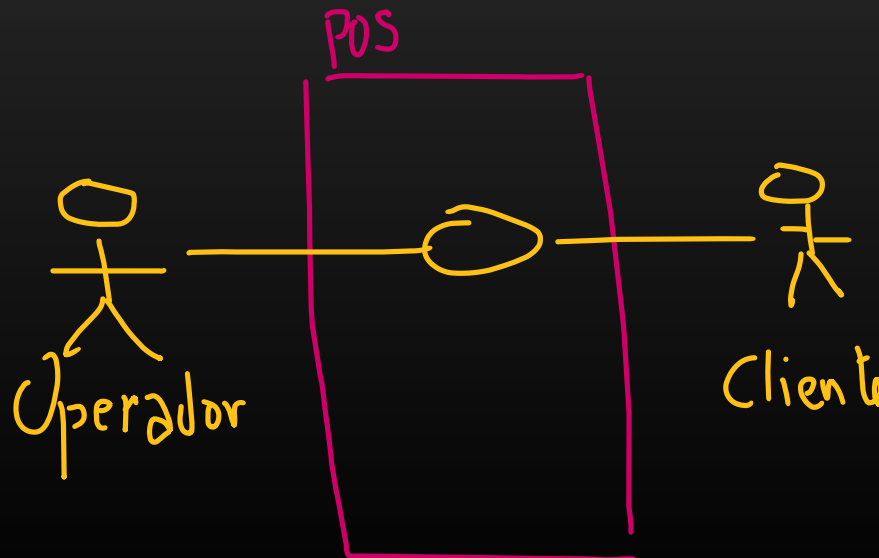
Solicita o sistema para resolver problemas/realizar objetivos

Os CaU são iniciados por um Ator primário

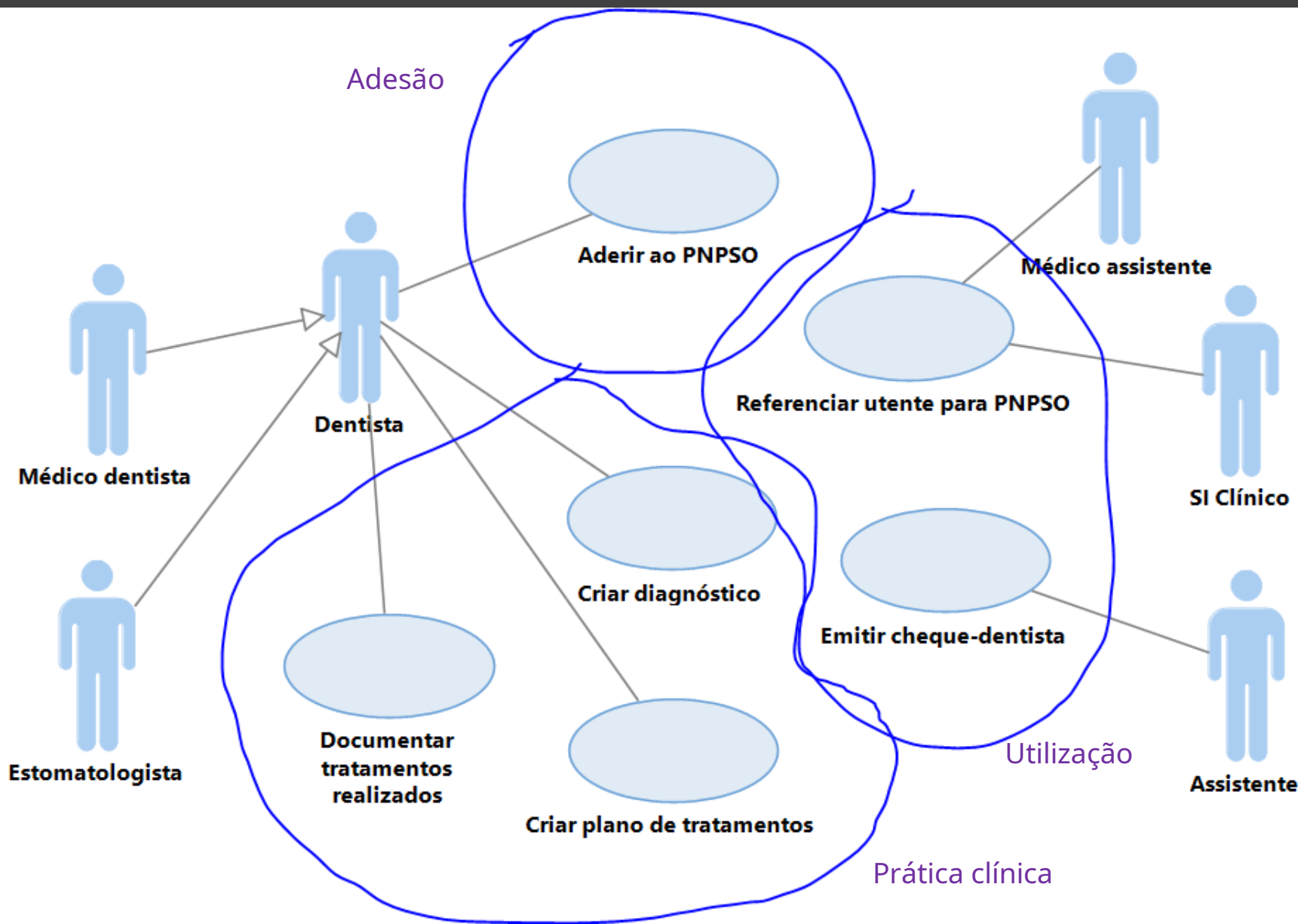
## Ator secundário

Fornece serviços ou informação para algum cenário do CaU

Podem ser sistemas externos ou papéis de pessoas, que não são utilizadores



## Podemos agrupar os casos de utilização em temas



## Os casos de utilização podem ser agrupados em pacotes

### general

- + Add to Wish List
- + Cancel Order
- + Edit Shopping Cart
- + Login
- + Logout
- + Open an Account
- + Return a book
- + View Order History
- + Where's My Stuff?

### admin

- + Customer Service
- + Seller
- + Shipping Clerk
- + Webmaster
- + Add Books to Catalog
- + Add Editorial Review
- + Add External Books to Catalog
- + Dispatch Order
- + Moderate Customer Reviews
- + Monitor Stock Levels
- + Order Books from Publisher
- + Process Refund
- + Remove Books from Catalog
- + Remove External Books from Catalog
- + Respond to Enquiry
- + Unlock Locked Account

### shopping

- + Customer
- + Add Item to Shopping Cart
- + Checkout
- + Edit Shopping Cart
- + Enter Address
- + Pay by Card
- + Pay by Check
- + Pay by Purchase Order
- + Remove Item From Shopping Cart
- + View Recommendations
- + View Review

### searching

- + Advanced Search
- + Search by Author
- + Search by Category
- + Search by Keyword
- + Search by Title
- + Search for Books

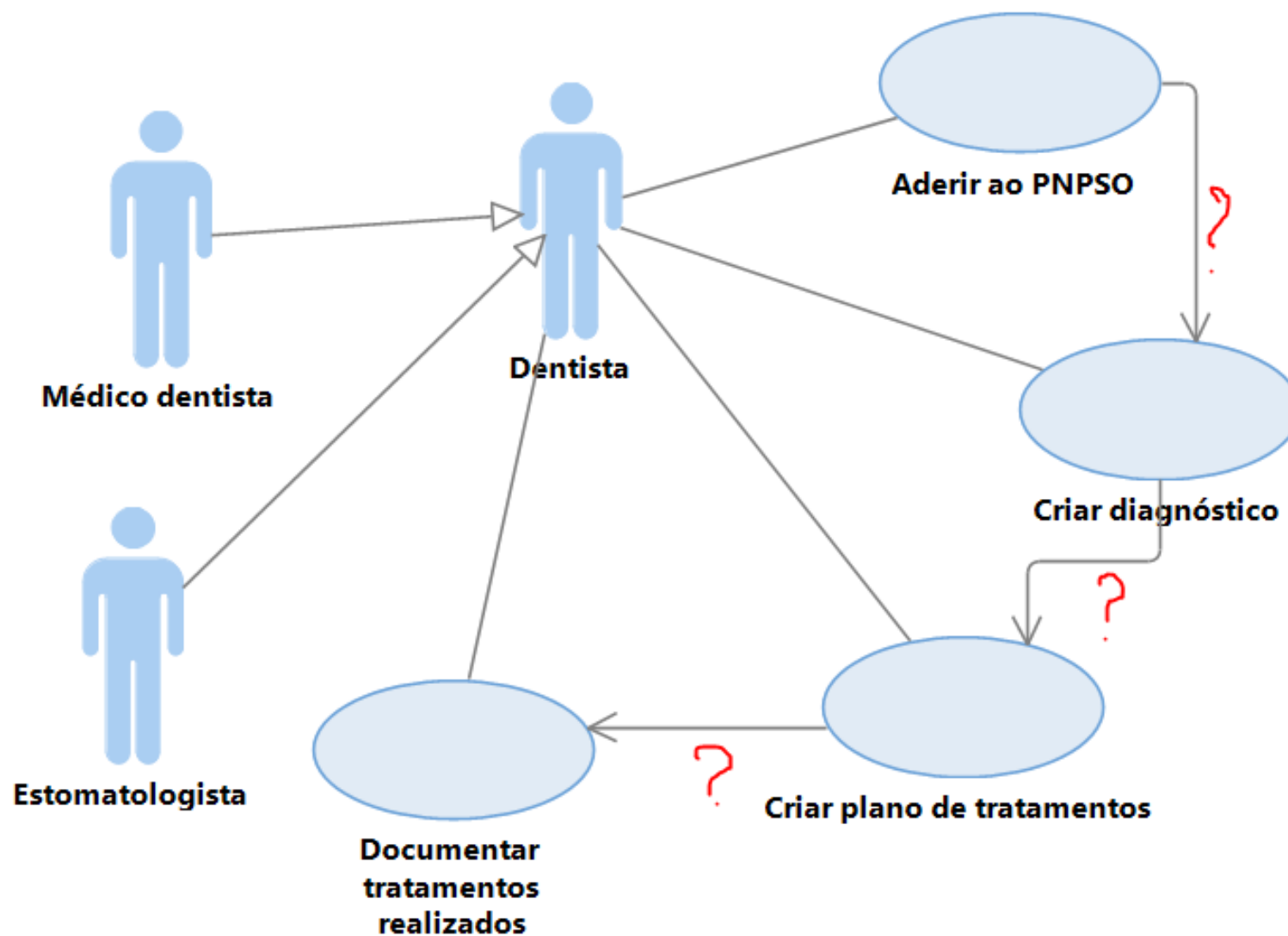
*Package*

*Actors*

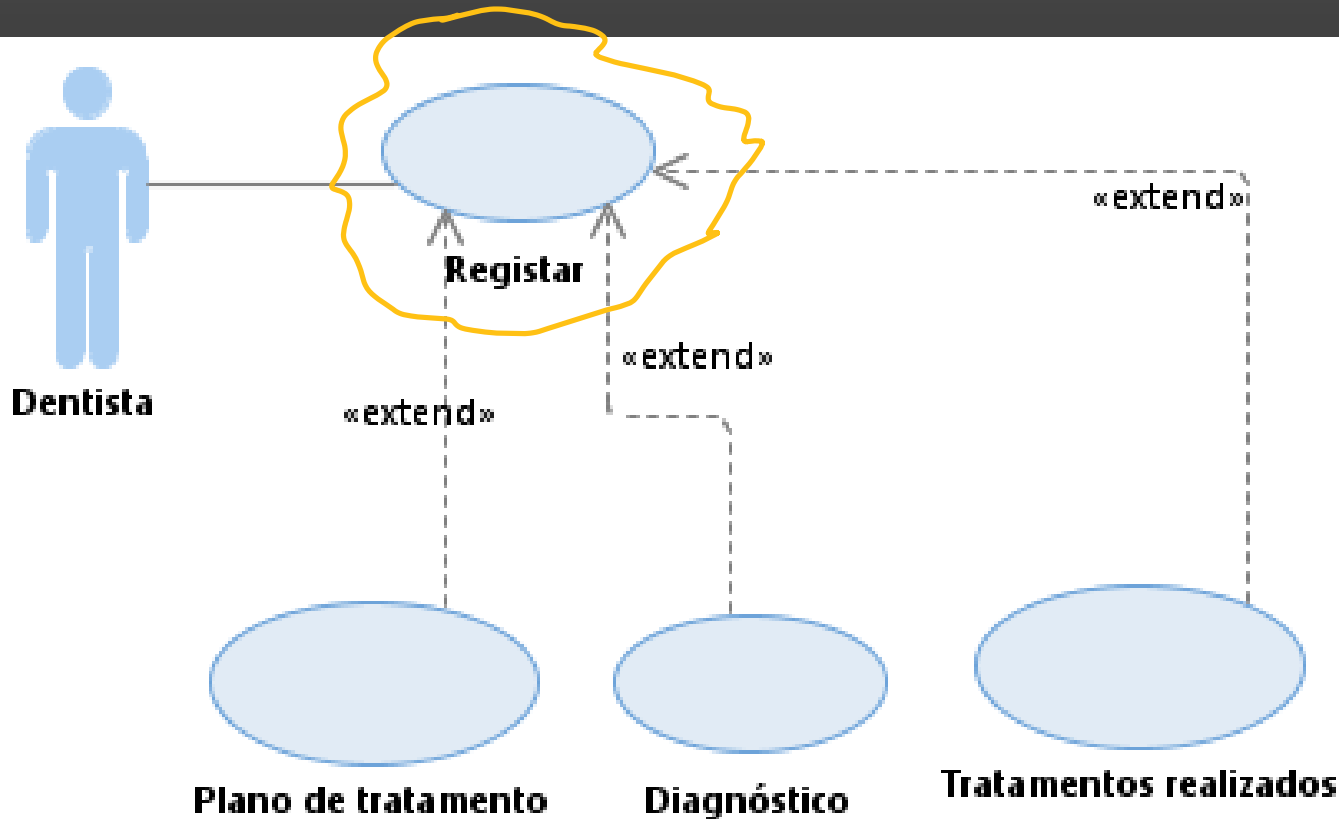
*Use cases*

As ferramentas de modelação permitem organizar o projeto em "models" e "packages"

## CaU não mostram *workflow*



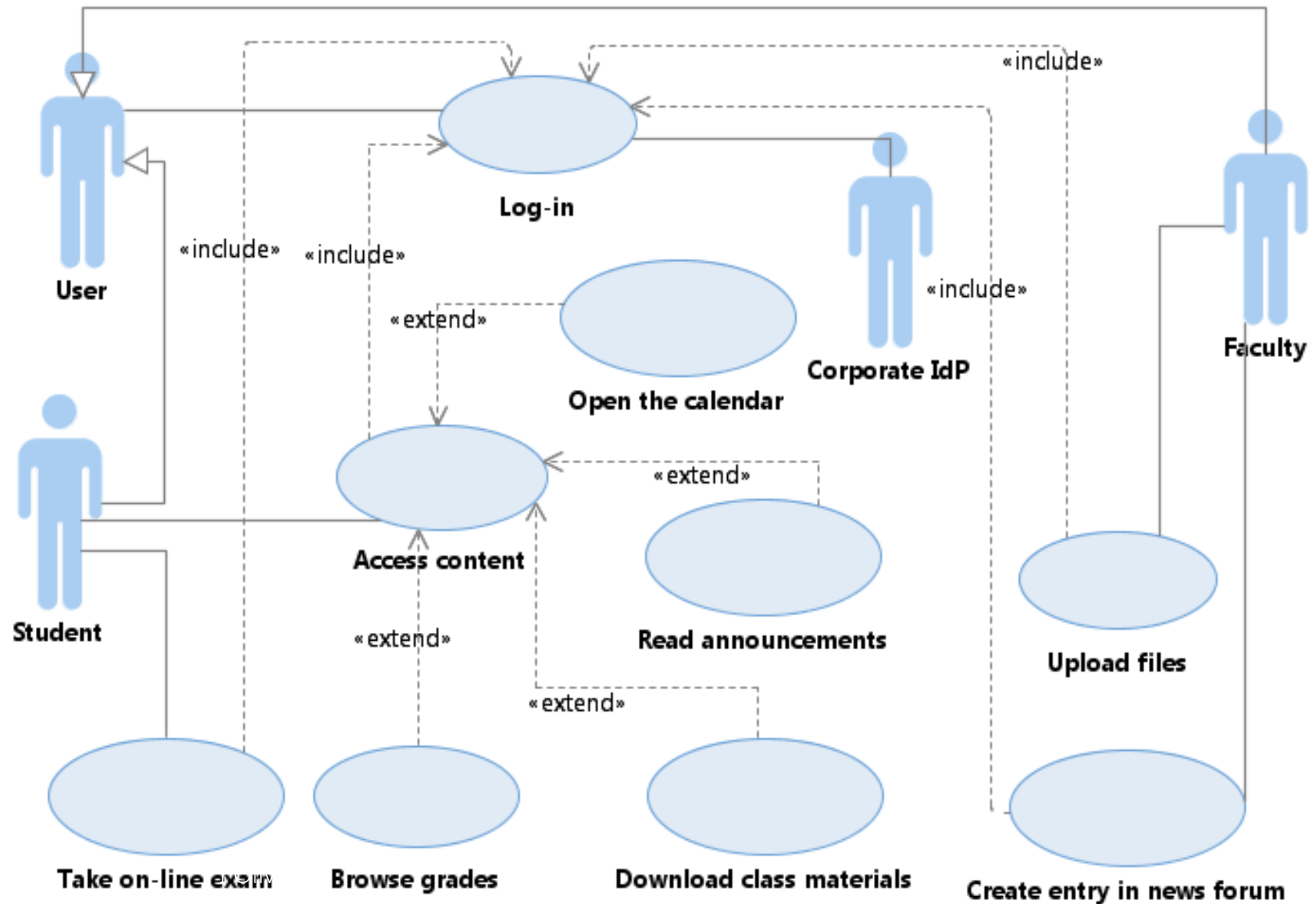
## Decomposição/Agregação funcional



Queremos manter a natureza de "episódios" de uso e mapear as motivações dos atores.

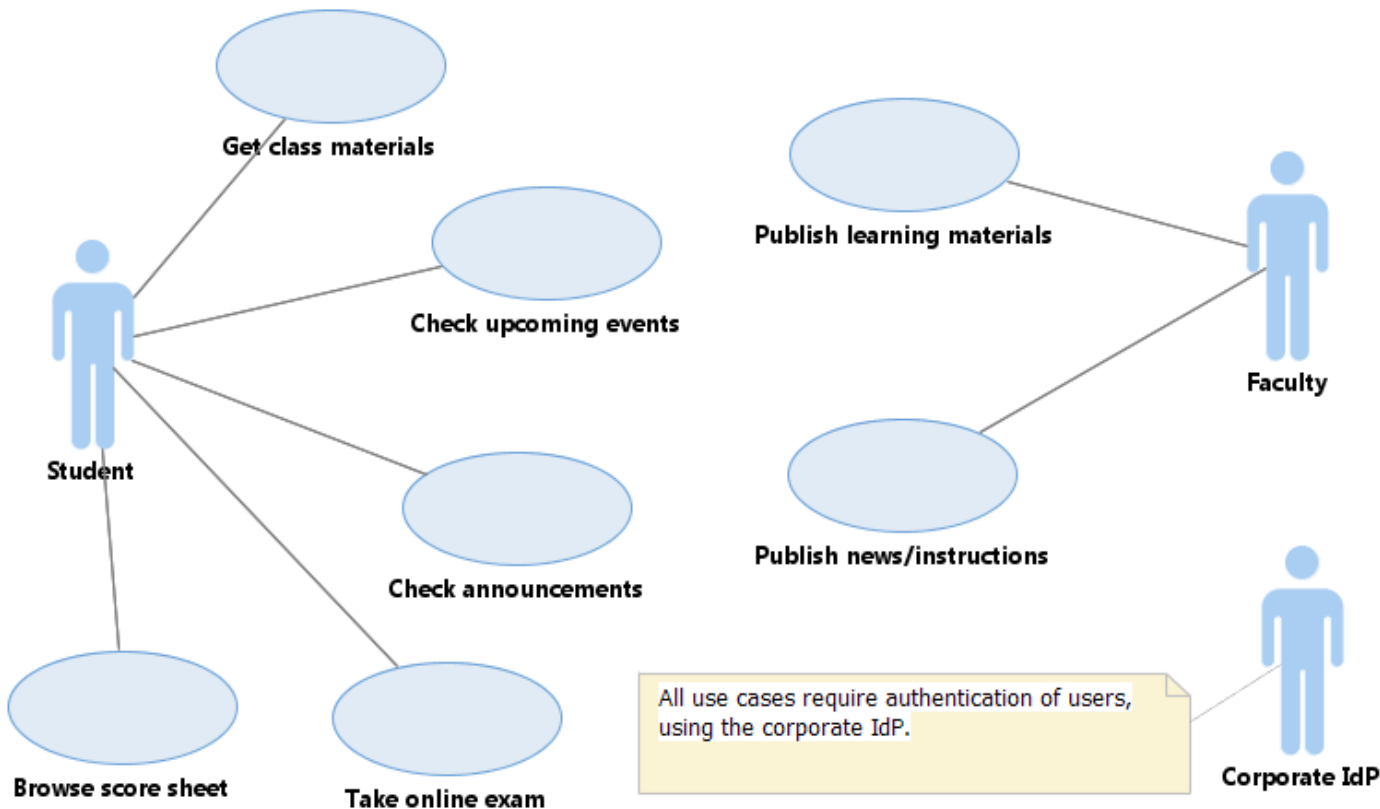
# Login e Logout como casos de utilização?

# Casos de utilização Moodle – Opção 1



## Casos de utilização Moodle – Opção 2

Evitar o uso de semântica avançada nos modelos, quando pode ser mais simples /direto.

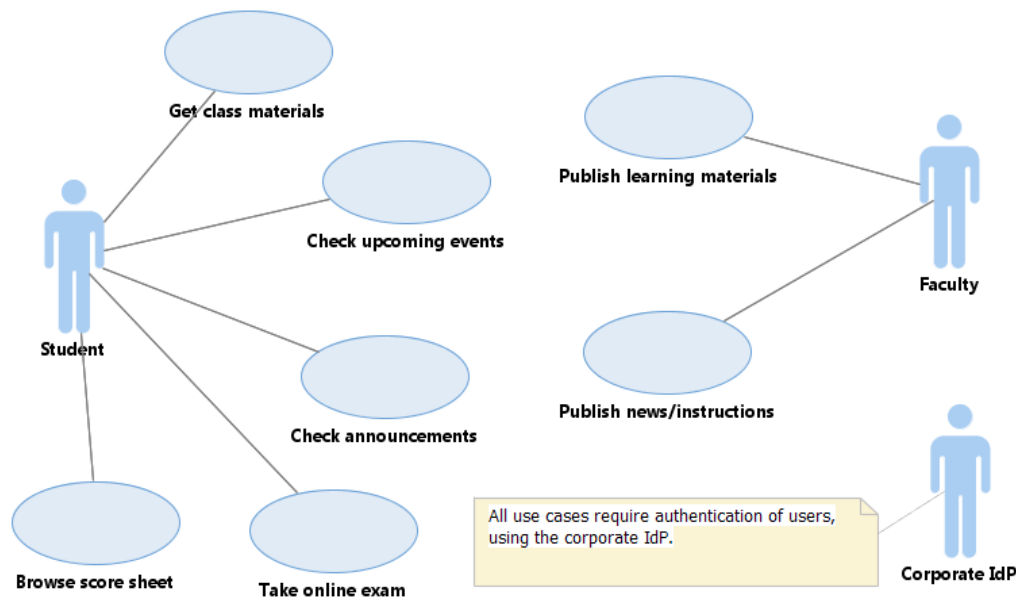
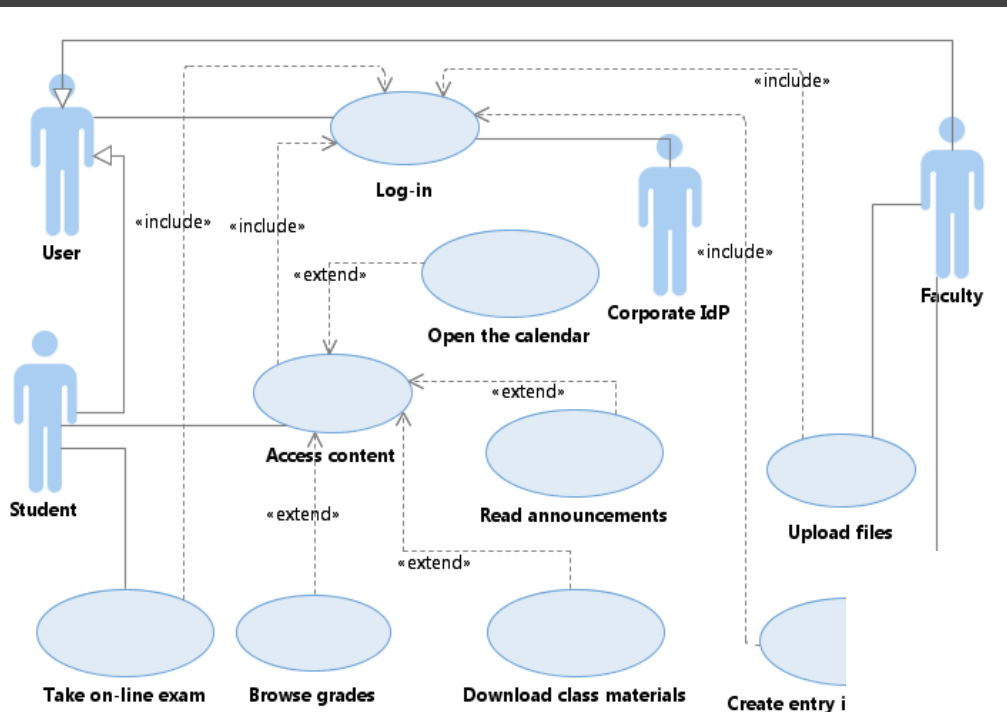




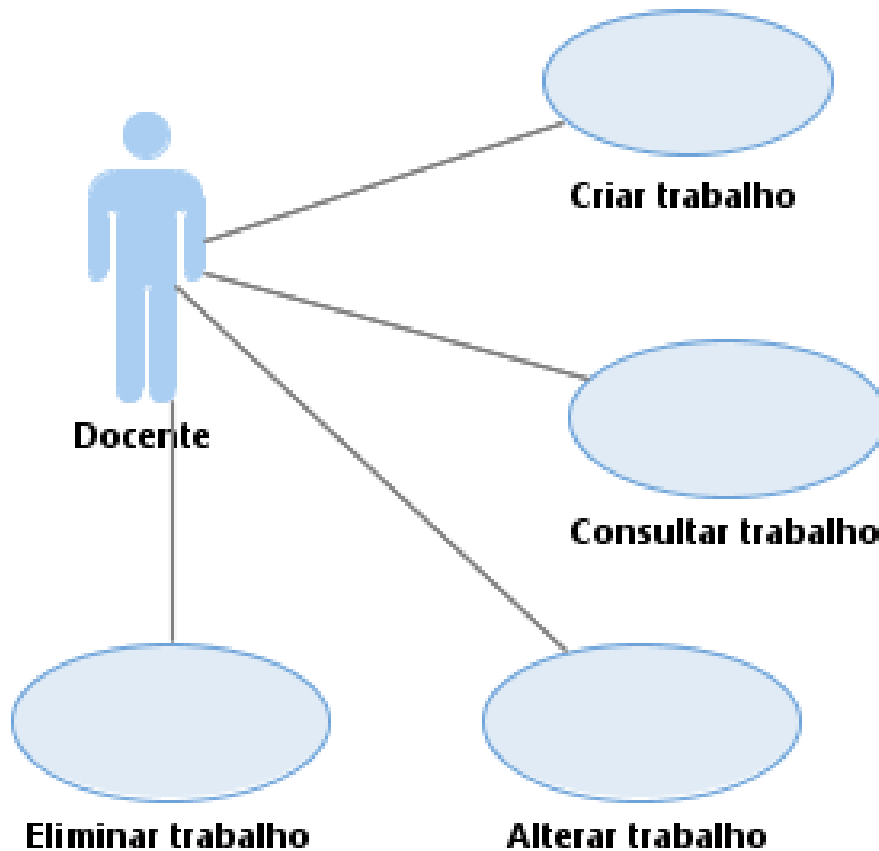
# Correção? Clareza? Eficácia?



Evitar o uso de semântica avançada nos modelos, quando se pode ser mais direto.



# Como lidar com o C.R.U.D. (create, retrieve, uppdate, delete)?



"objeto" do domínio: um trabalho (da disciplina)



"objeto" do domínio: um pedido (encomenda de comida)

Temos de analisar a "visibilidade" destes episódios na motivação dos atores

# Como lidar com o CRUD?

## Depende do problema!

Cancelar Cheque-dentista

Cancelar encomenda

Cancelar trabalho (Moodle)

+

+/-

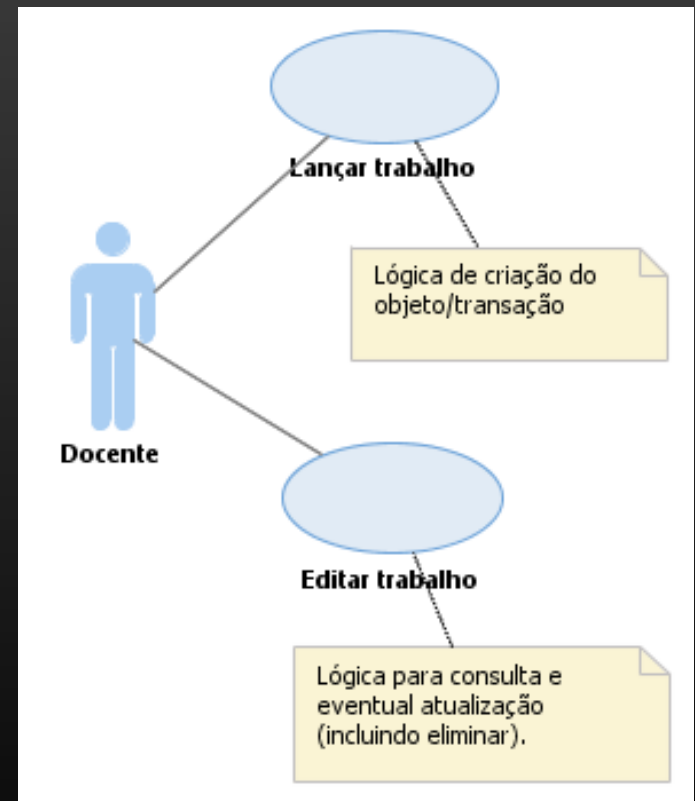
-

## Frequentemente, pode ser resolvido com dois CaU

O fluxo que cria (e.g.: Lançar trabalho, Pedir menu, Referenciar para cuidados de saúde oral,...)

O fluxo que consulta/atualiza/apaga

Em geral, **não** queremos modelar um CaU por cada situação CRUD relativa a um objeto / conceito do domínio

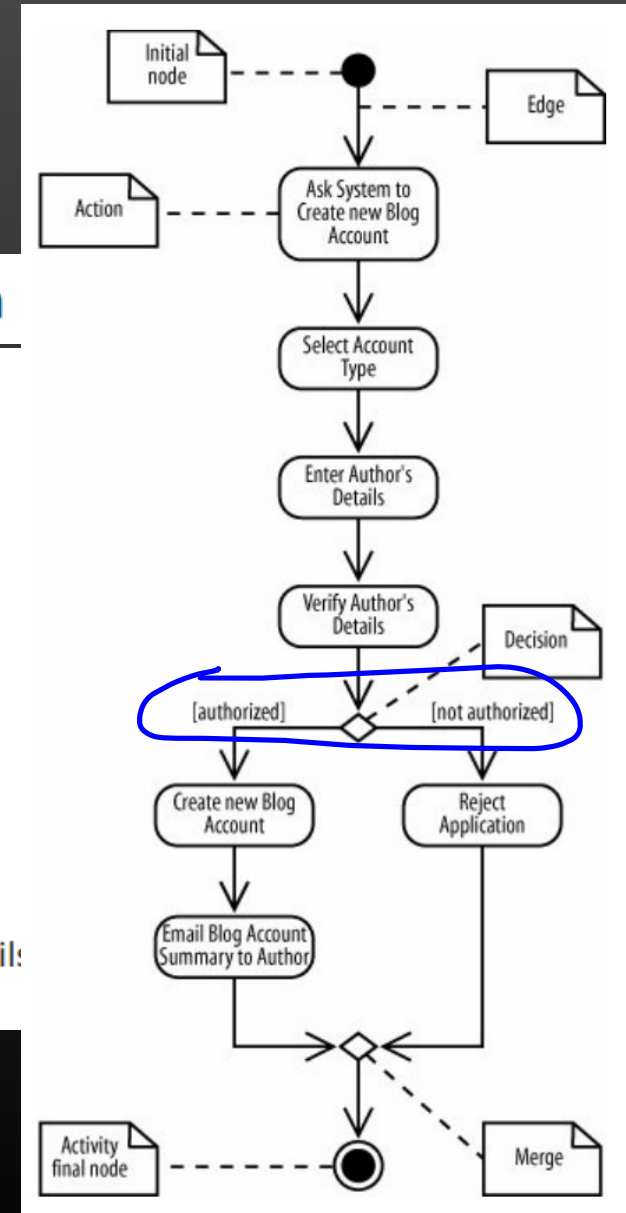


# Explicar os fluxos com DA

## Create a new Blog Account use case description

Step	Action	Atores como partições
1	The Administrator asks the system to create a new blog account.	
2	The Administrator selects an account type.	
3	The Administrator enters the author's details.	
4	The author's details are verified using the Author Credentials Database.	
5	The new blog account is created.	
6	A summary of the new blog account's details are emailed to the author.	
Step	Branching Action	
4.1	The Author Credentials Database does not verify the author's details:	

Relevante quando os cenários são elaborados e/ou com várias "responsabilidades"



## Recapitulando: para que serve o modelo de casos de utilização?

Uma vista de um sistema que destaca o comportamento observável, tal como é percebido pelos utilizadores

O modelo de casos de utilização divide a funcionalidade do sistema em episódios relevantes para os utilizadores/atores

Capta os problemas/motivações que levam à utilização do sistema

Capta a funcionalidade do sistema (“o quê”), não a implementação da solução (“o como”)

A UML fornece uma visualização, mas o mais completo é a narrativa!

# Readings & references

Core readings	Suggested readings
<ul style="list-style-type: none"><li>• [Dennis15] – Chap. 3 – Requirements Determination</li></ul>	<ul style="list-style-type: none"><li>• [Pressman15] – Chap. 8 – Understanding Requirements</li><li>• [Wiegers13] – Chap. 1 -3</li><li>• [Larman 12] - Chap. 5</li></ul>