47006- ANÁLISE E MODELAÇÃO DE SISTEMAS

Práticas da Análise no projeto de ASis

Ilídio Oliveira

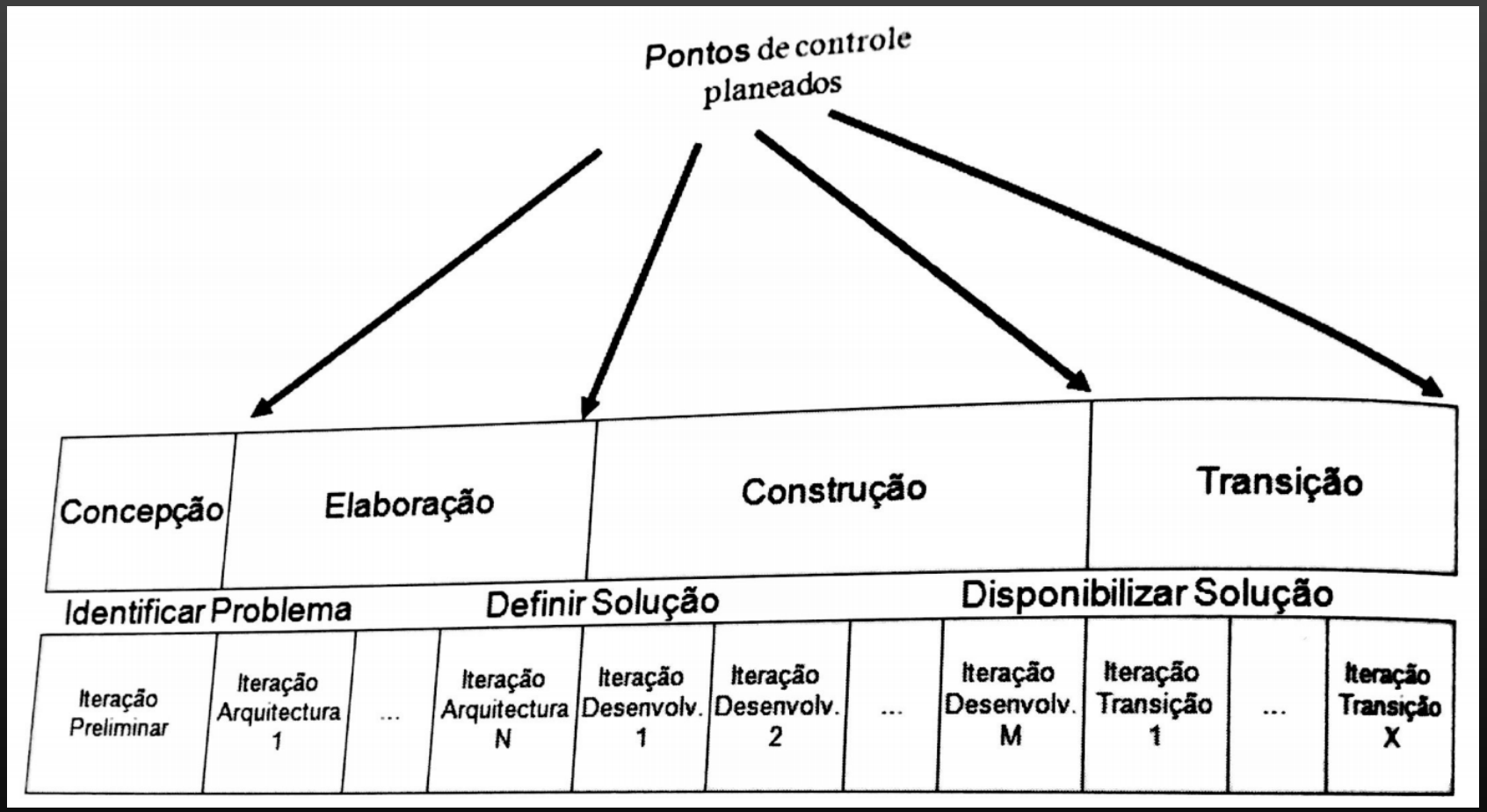
V2022/05/10

universidade de aveiro
departamento de eletrónica,
telecomunicações e informática
deti

Objetivos de aprendizagem

- Explicar as próximas atividades no desenvolvimento do projeto em grupo.
- Identificar os resultados produzidos pelos Analista na investigação do domínio (relacionado com o OpenUP).

OpenUP: Fases, iterações e pontos de controlo



Novos processos

Reengenharia de processos

Analizar a forma de trabalhar da organização

- Novo conceito *(green-field)*?
- Reegenharia?
- Ainda não é uma especificação de software!

Procurando oportunidades de melhoria

- "gerar valor"
- Processos mais rápidos, mais convenientes, novos serviços,...

Potenciadas pelas TI

https://www.gartner.com/en/information-technology/insights/top-technology-trends

Elementos de um processo

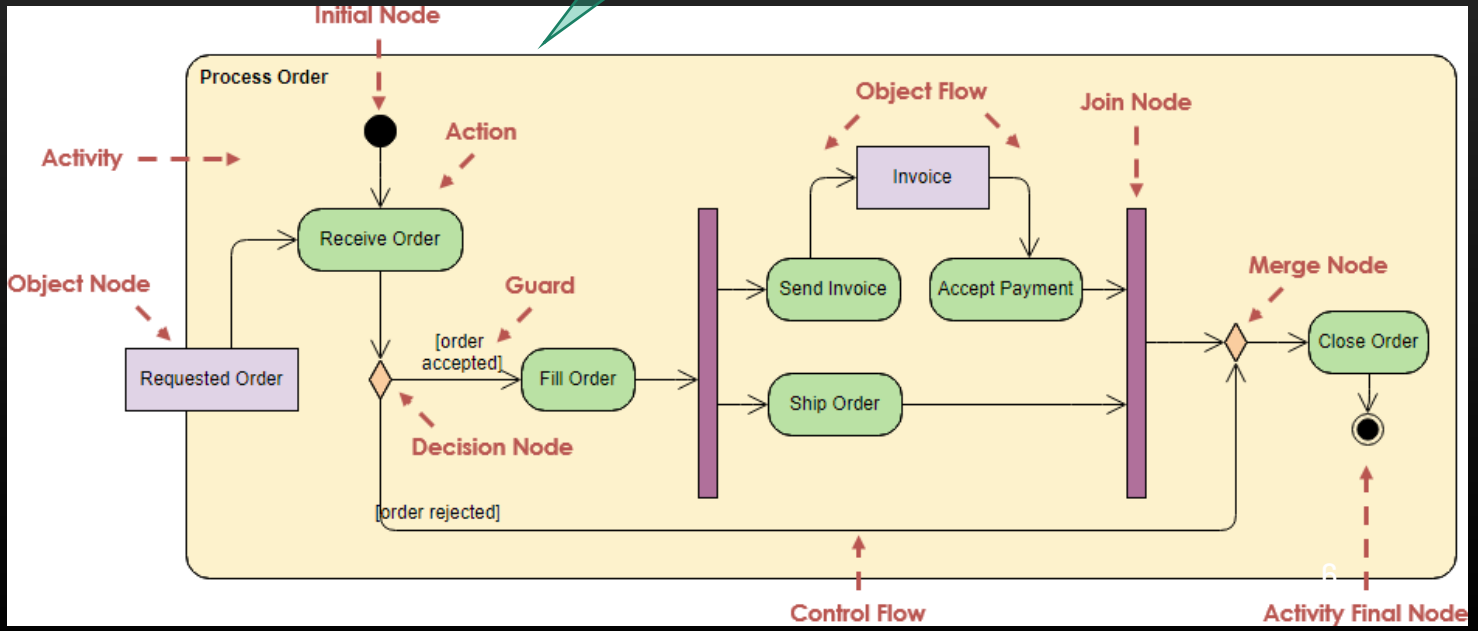
Intervenientes/papéis

Tarefas/"Ações" & Eventos

Fluxo/Decisão

Inputs/Outputs (informação)

O Analista pode usar D. Atividades para documentar os futuros processos



Problemas mais frequentes

Um (único) processo (!)

- Vs: Compra online,
  Monitorização da entrega,
  Gestão de parcerias,...

Detalhe excessivo para o objetivo

- clarificar as responsabilidade e fluxo no novo “conceito”

Pouco estudo do domínio...

Focar num caso particular e não no domínio/área

Modelo [dos conceitos] do domínio

Mapa de conceitos

Os conceitos de um problema tendem a manter-se estáveis

Input para:

- Vocabulário coerente nas especificações
- Bases de dados
- “Domain-driven design”
- Podem revelar *Business rules*

Problemas mais frequentes

Informação a menos

Faltam entidades, faltam atributos, não responde ao prometido nos CaU

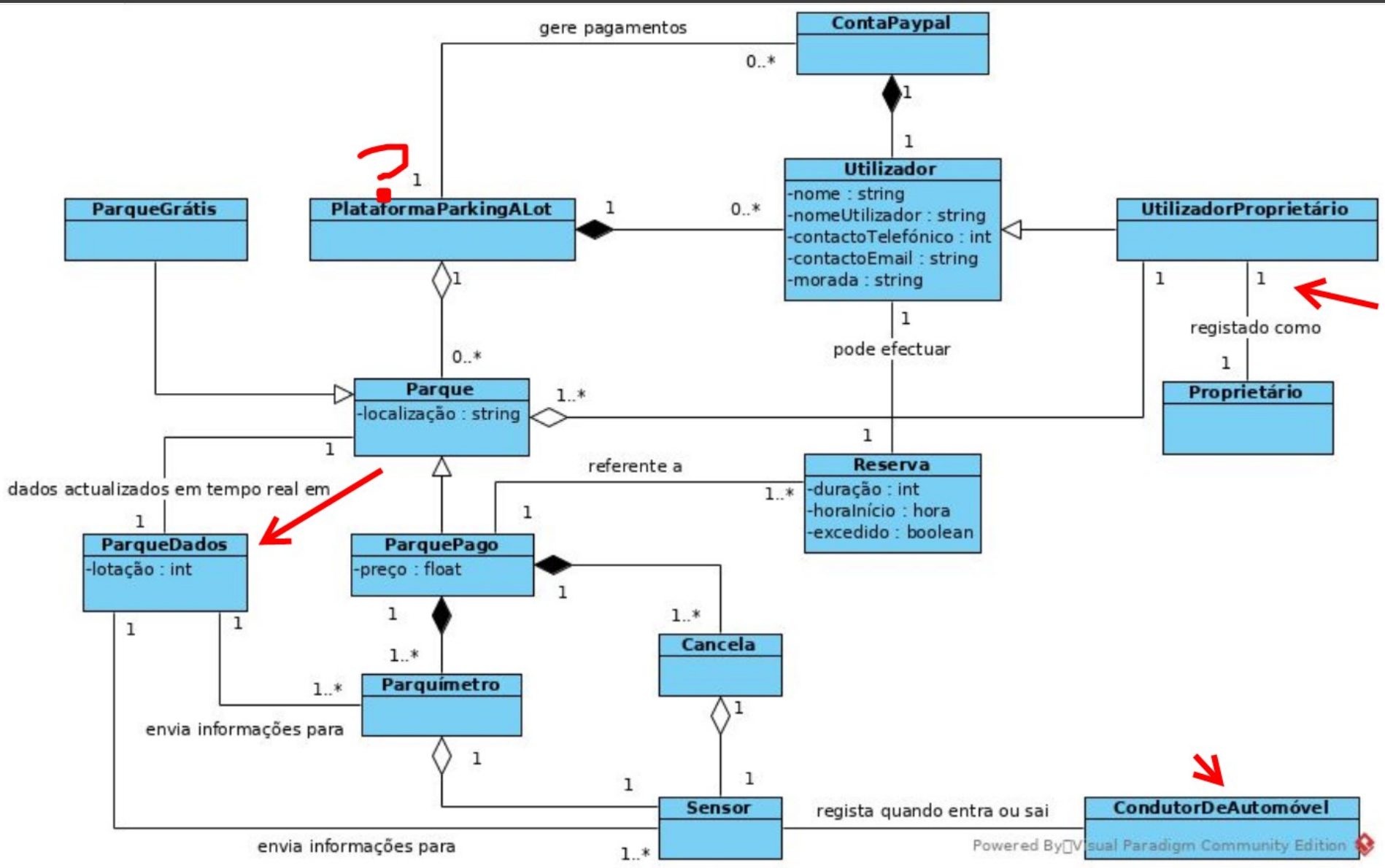
Tempo

Consigo reproduzir a informação de há 6 meses? Consigo produzir "mapas mensais"?

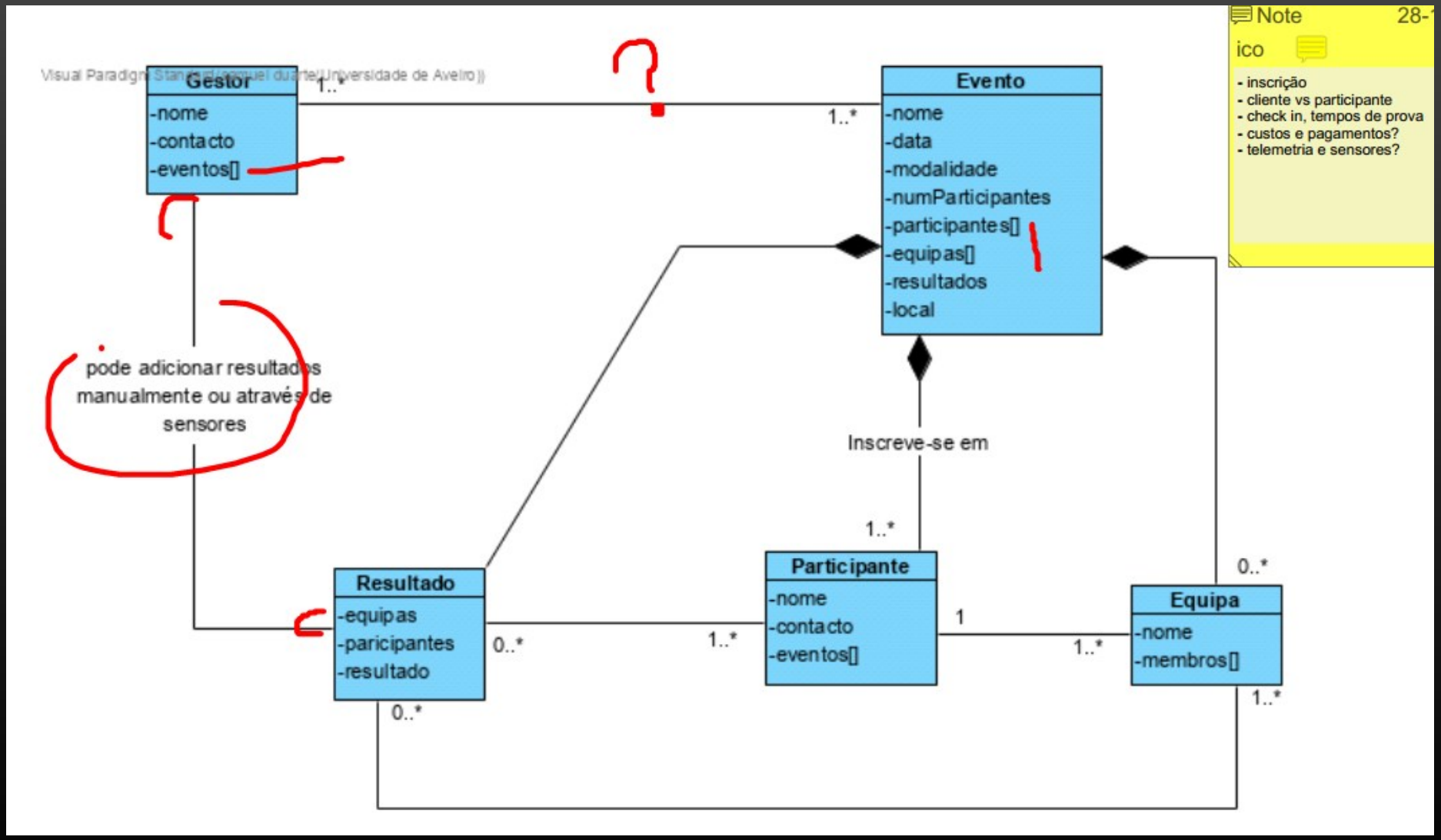
Detalhes da implementação

Login?...

Exemplo para discussão (gestão de parques)



Exemplo p/ discussão (gestão de corridas)



Informação a diferentes níveis

A componente operacional está assegurada?

→ Suporte às transações do dia-a-dia

A componente tática está assegurada?

→ Informação para a camada (intermedia) de gestão do negócio

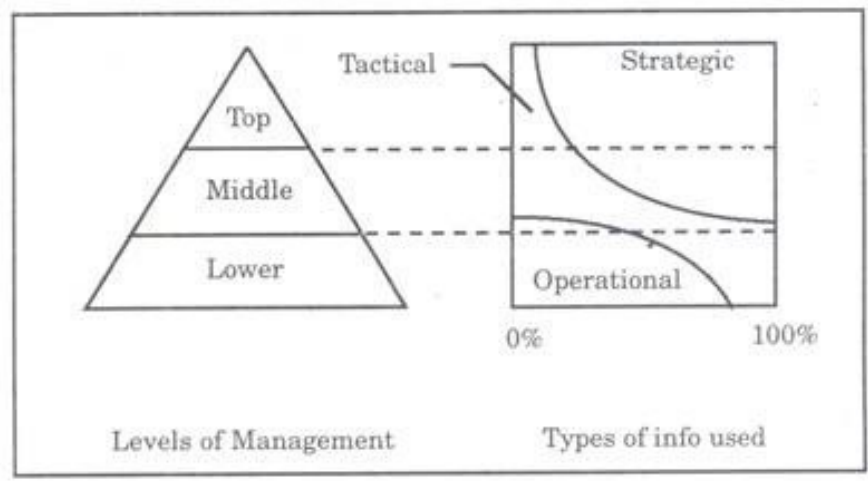


Fig. 1.2 Types of information and levels of management

O "big-boss" como um *stakeholder*...

Papel dos casos de uso

Posicionamento do UP

Segundo os autores:

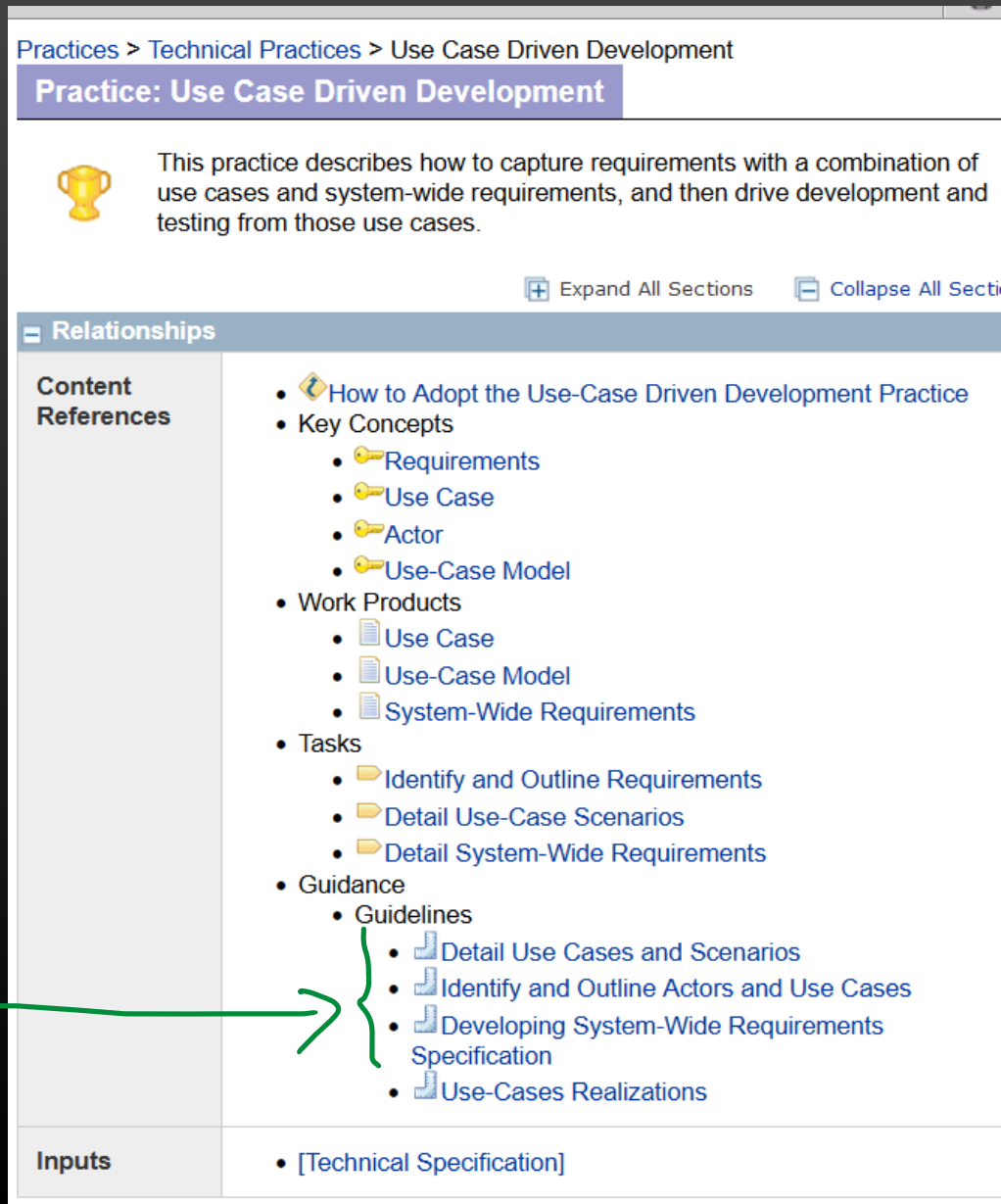
“architecture centric”

“use-case driven”

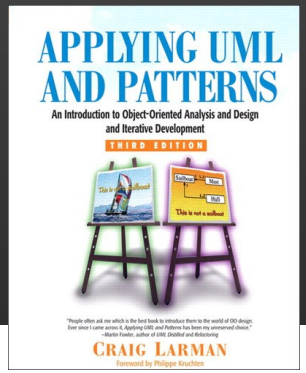
“interactive and incremental”

Desenvolvimento por casos de utilização

"This practice also explains how use cases and scenarios are best developed in conjunction with (and used to drive) other development activities, including design and testing."



Relevância para a análise de requisitos



The Boss Test

Your boss asks, “What have you been doing all day?” You reply: “Logging in!” Is your boss happy?

If not, the use case fails the Boss Test, which implies it is not strongly related to achieving results of measurable value. It may be a use case at some low goal level, but not the desirable level of focus for requirements analysis.

That doesn't mean to always ignore boss-test-failing use cases. User authentication may fail the boss test, but may be important and difficult.

E os requisitos não funcionais

“System-wide specification”

Situações de modelação com CaU

universidade de aveiro
departamento de eletrónica,
telecomunicações e informática

deti

USE CASE 24: FULLY DRESSED USE CASE TEMPLATE <NAME>

<the name should be the goal as a short active verb phrase>

Context of use: *<a longer statement of the goal, if needed, its normal occurrence conditions>*

Scope: *<design scope, what system is being considered black-box under design>*

Level: *<one of: summary, user-goal, subfunction>*

Primary Actor: *<a role name for the primary actor, or description>*

Stakeholders & Interests: *<list of stakeholders and key interests in the use case>*

Precondition: *<what we expect is already the state of the world>*

Minimal Guarantees: *<how the interests are protected under all exits>*

Success Guarantees: *<the state of the world if goal succee*

Trigger: *<what starts the use case, may be time event>*

Main Success Scenario:

<put here the steps of the scenario from trigger to goal del

<step #> <action description>

Extensions:

<put here there [sic] *extensions, one at a time, each referring to the step of the main scenario>*

<step altered> <condition>: <action or sub use case>

<step altered> <condition>: <action or sub use case>

Technology & Data Variations List:

<put here the variations that will cause eventual bifurcation in the scenario>

Start simple, then refine.

Avoid "overloaded" use-case specification templates

Atores primários e secundários

Ator primário

Solicita o sistema para resolver problemas/realizar objetivos

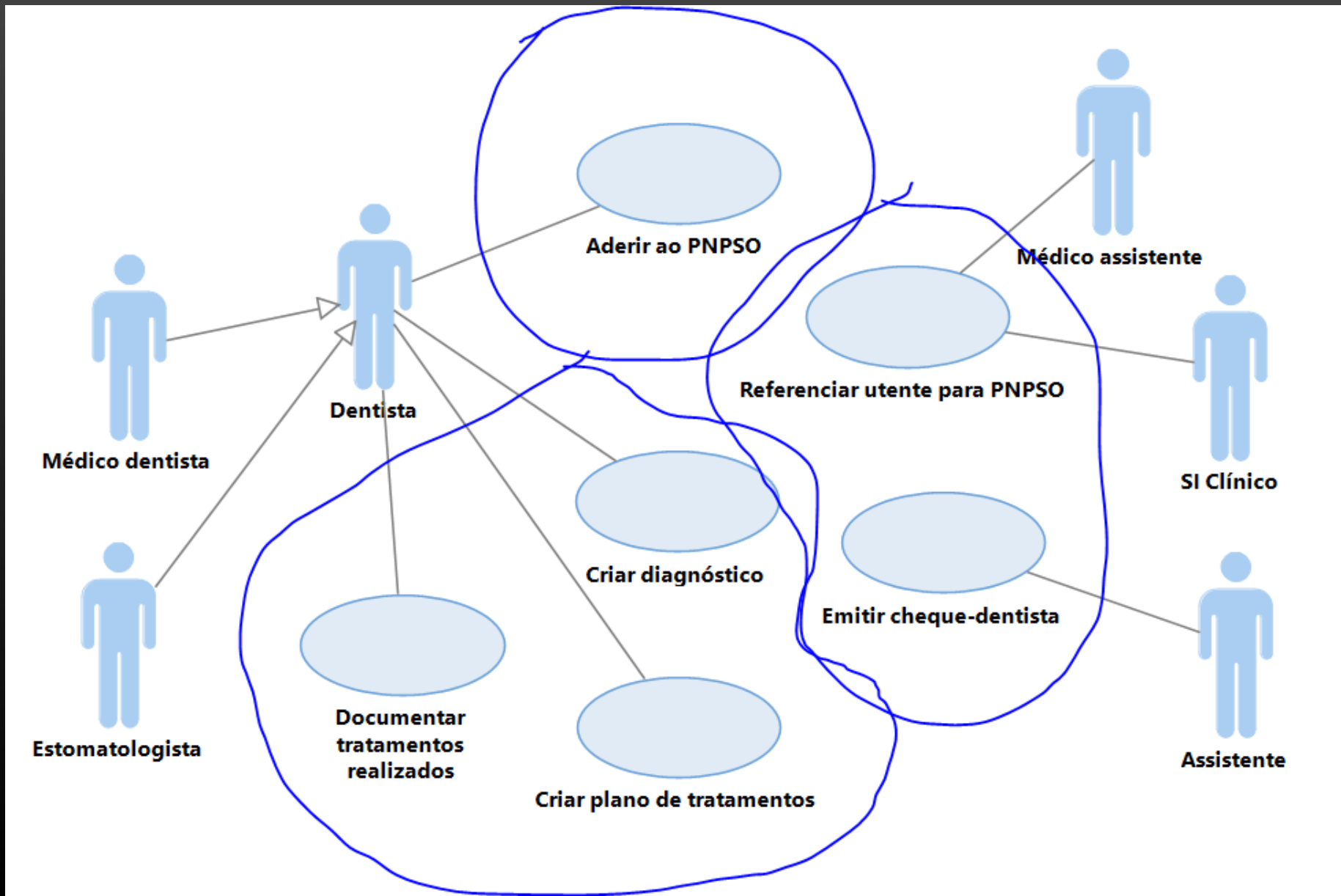
Os CaU são iniciados por um Ator primário

Ator secundário

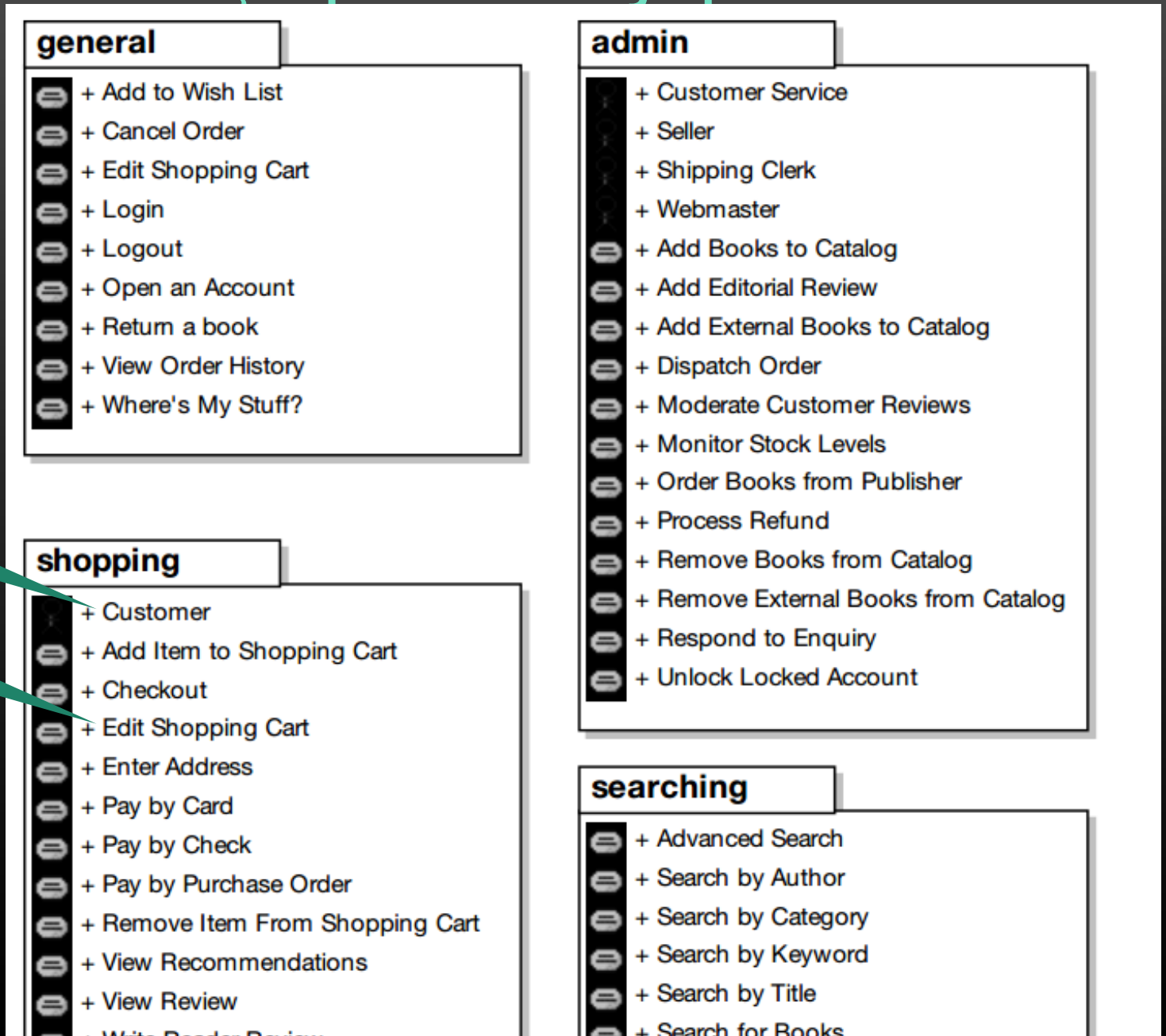
Fornece serviços ou informação para algum cenário do CaU

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

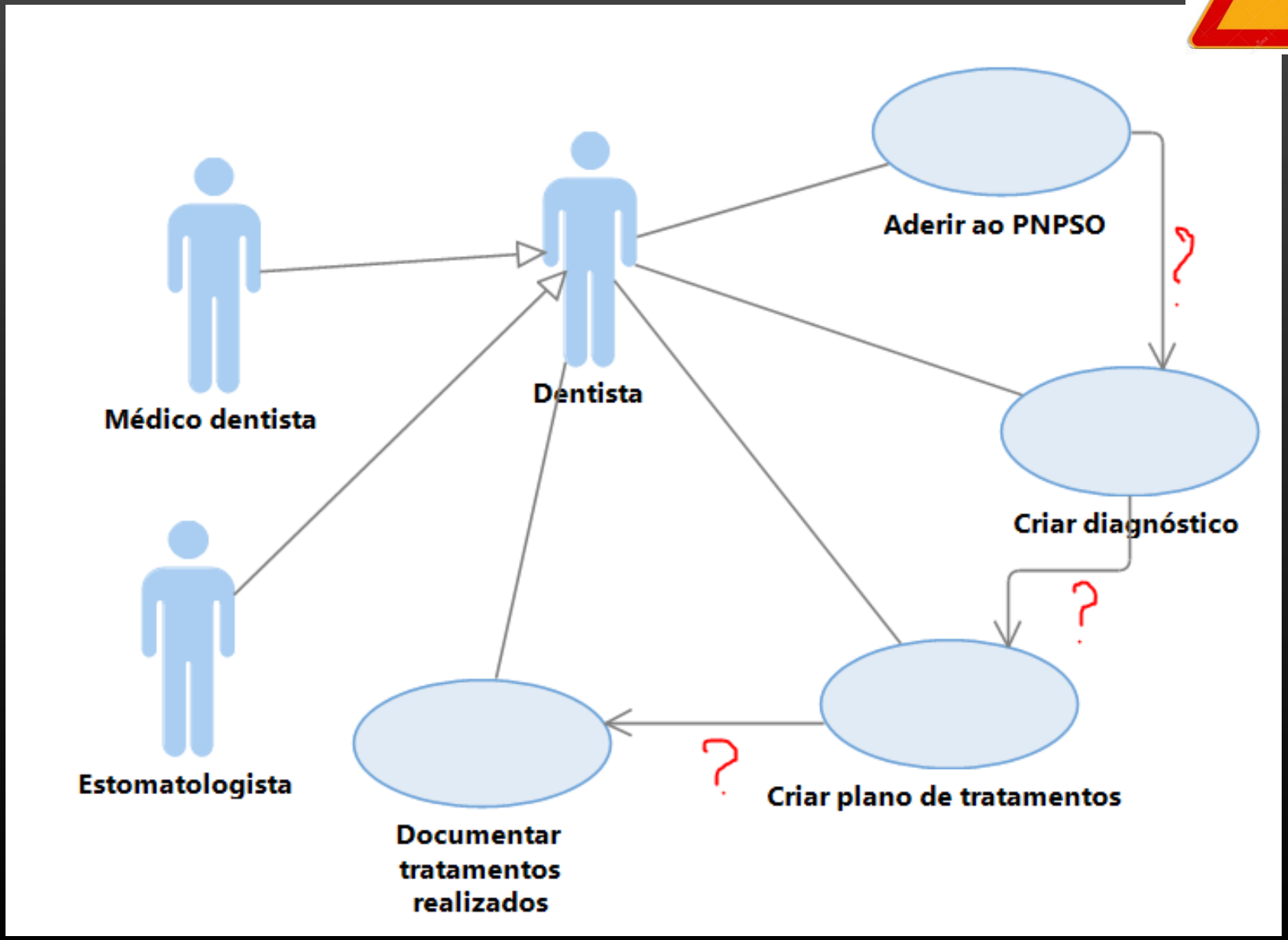
Podemos agrupar os casos de utilização em temas



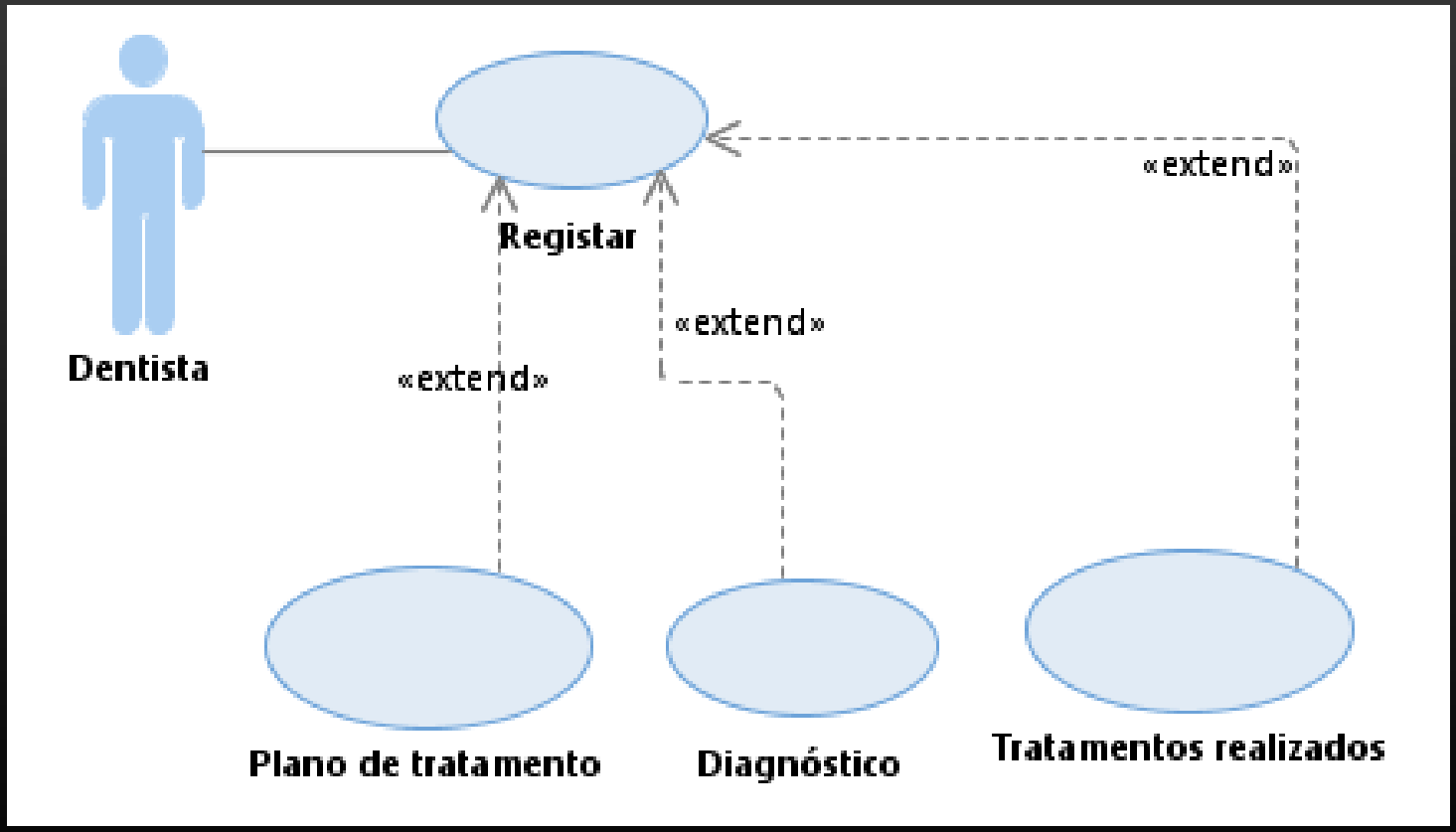
Os casos de utilização podem ser agrupados em pacotes



CaU não mostram *workflow*

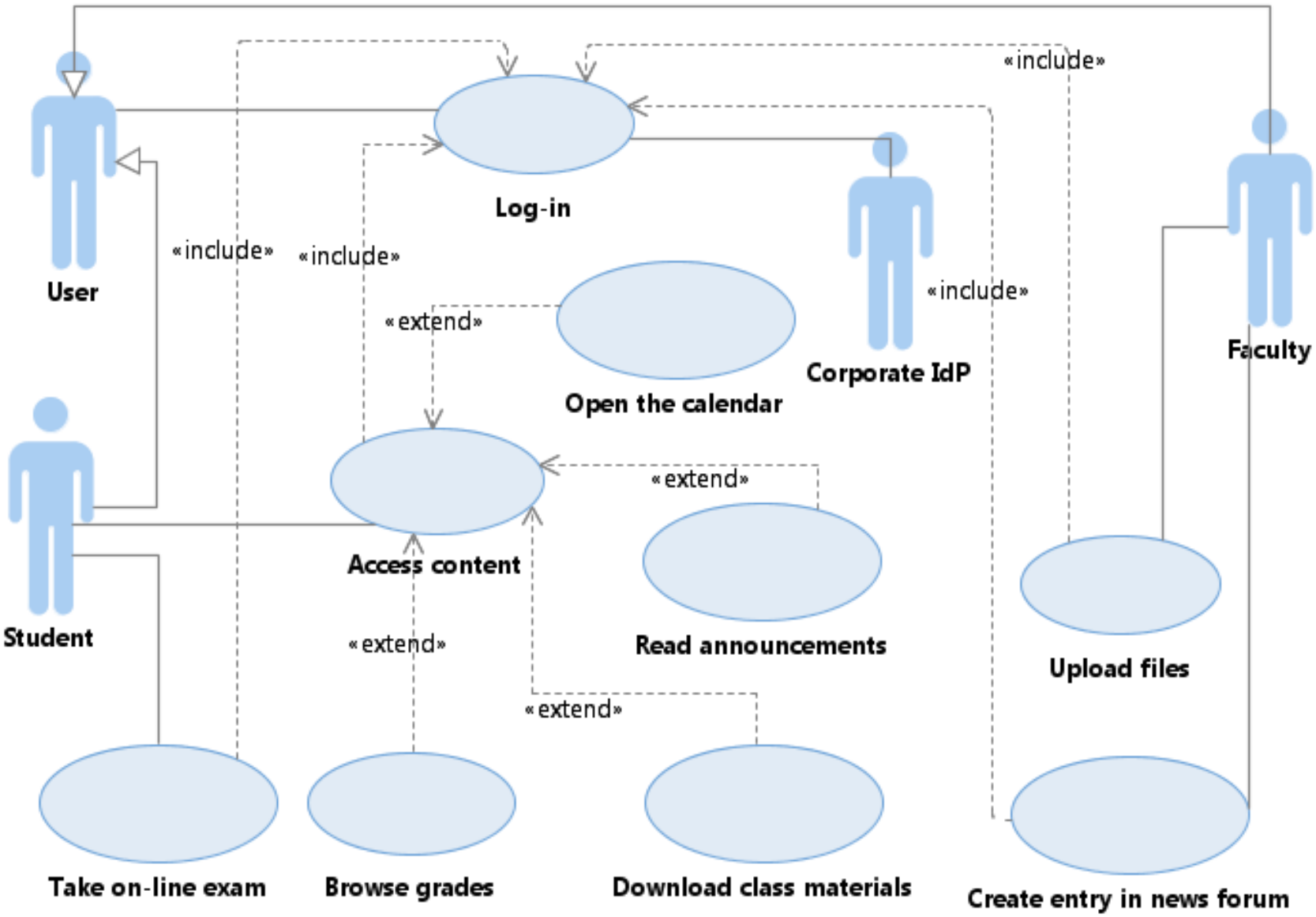


Decomposição/Agregação funcional

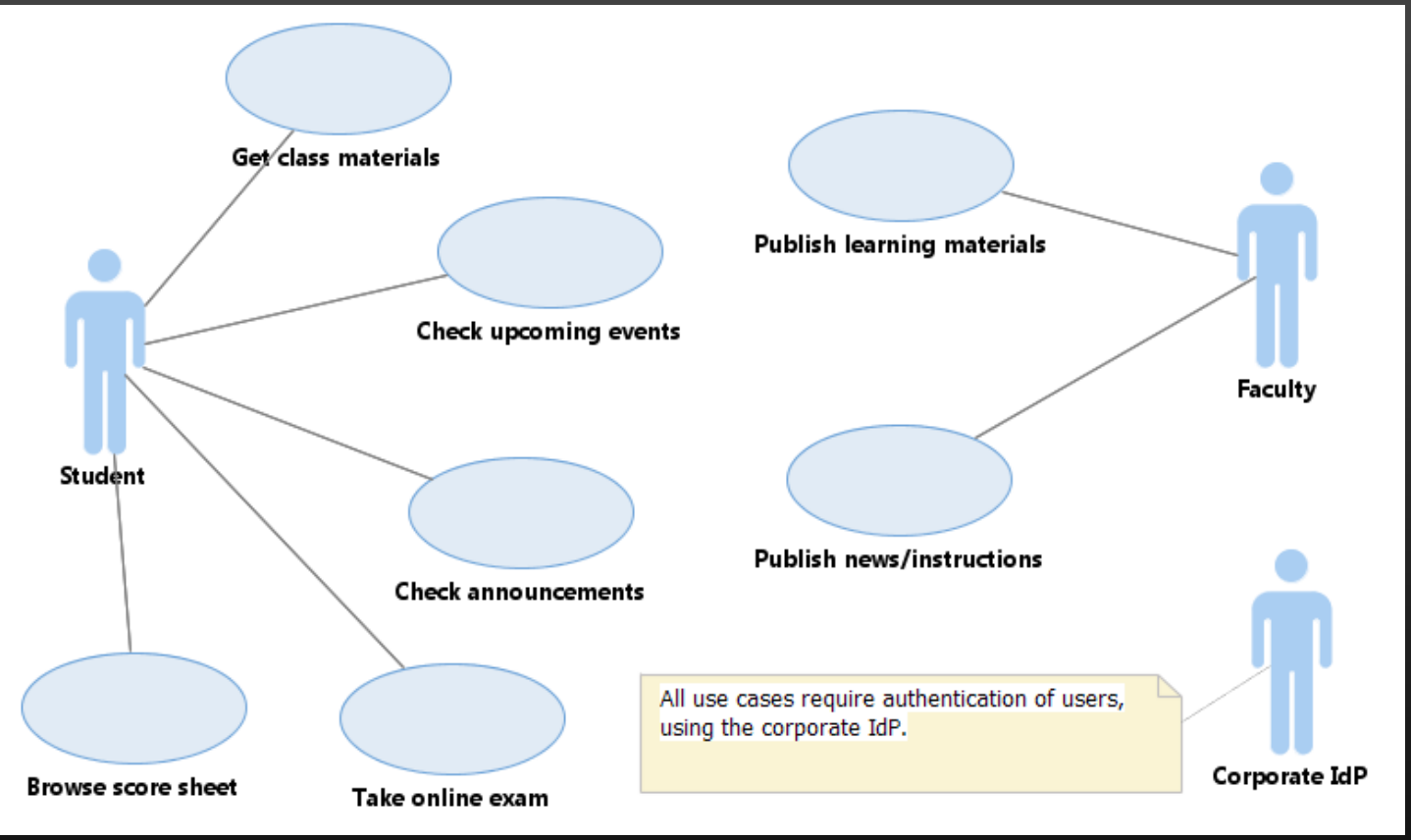


Login e Logout são casos de utilização?

Casos de utilização Moodle – Opção 1

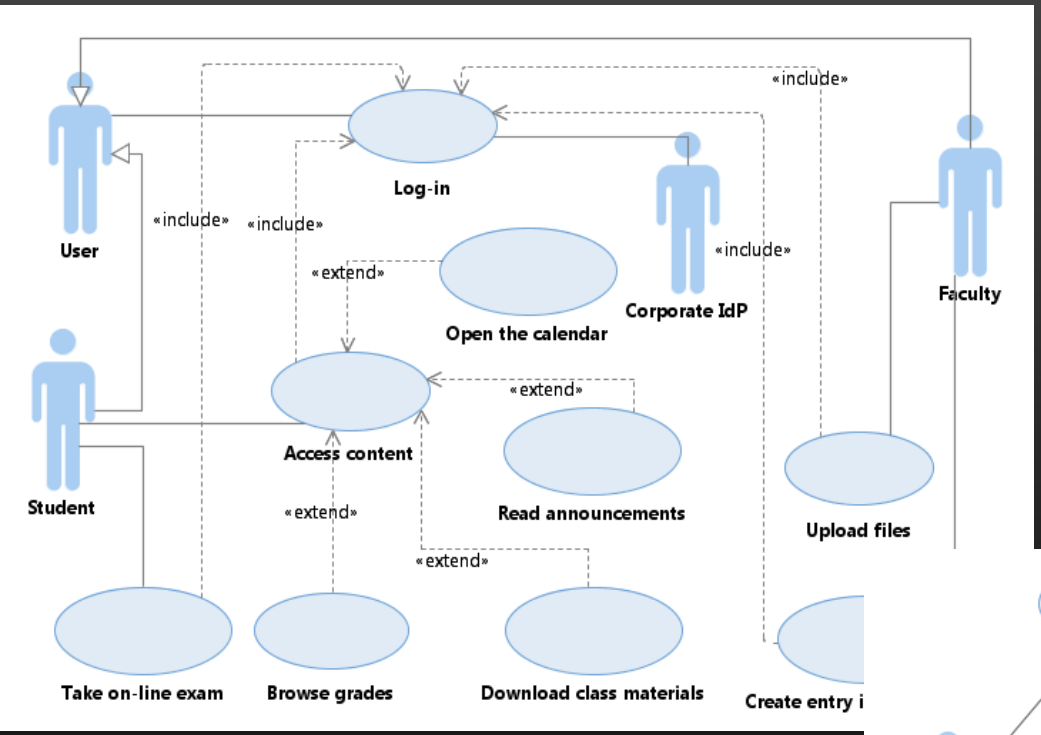


Casos de utilização Moodle – Opção 2

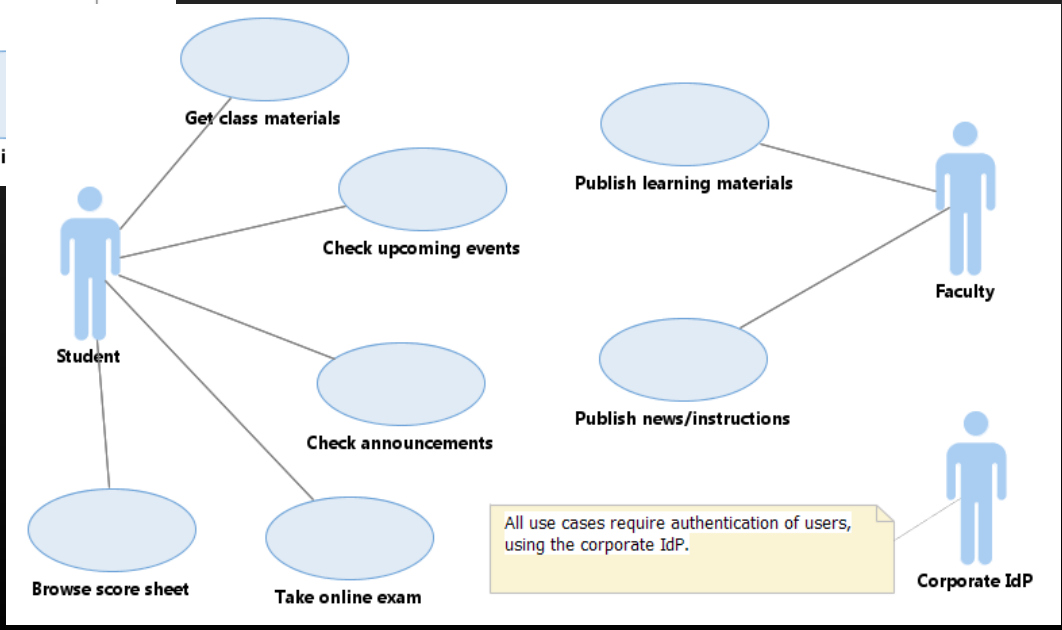


Evitar o uso de semântica avançada nos modelos, quando pode ser mais simples /direto.

Correção? Clareza? Eficácia?

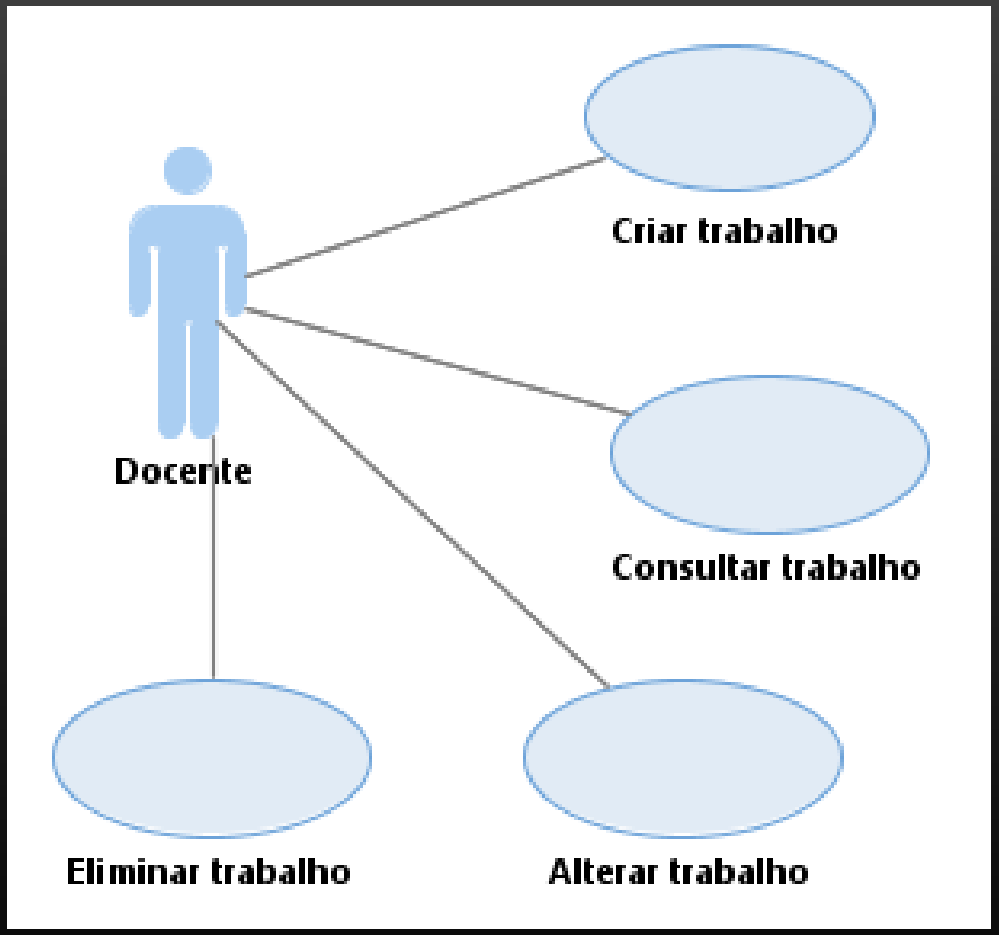


Evitar o uso de semântica avançada nos modelos, quando pode ser mais simples /direto.



I Oliveira (2018)

Como lidar com o CRUD?



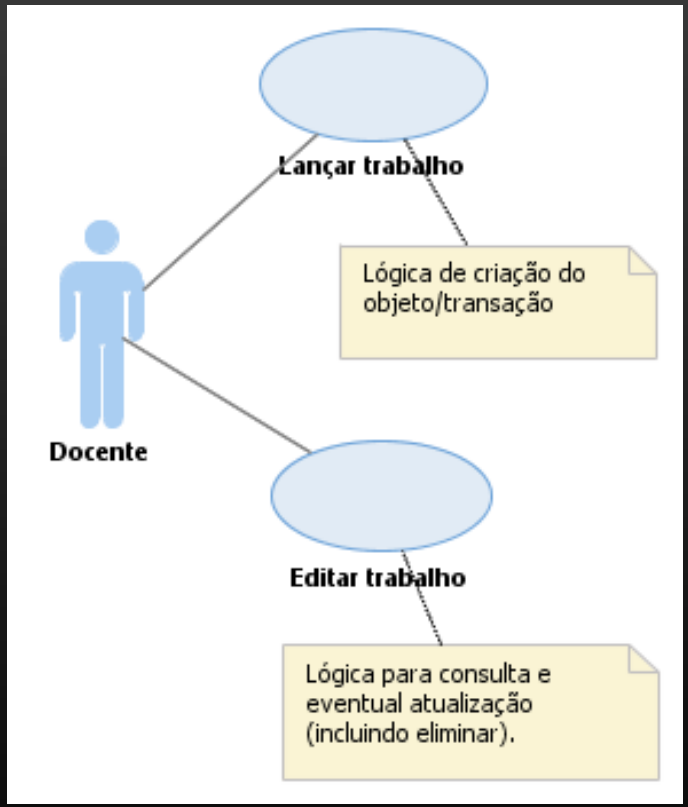
Como lidar com o CRUD?

Depende do problema

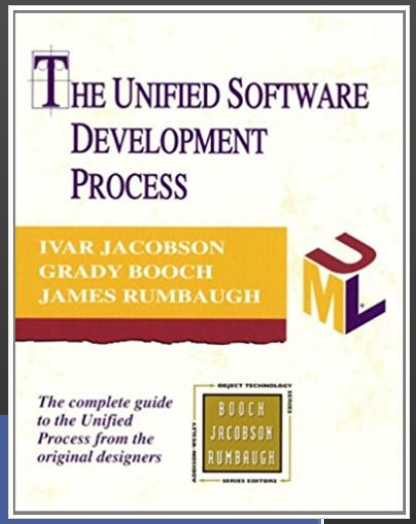
- Cancelar Cheque-dentista vs Cancelar trabalho (Moodle)

Frequentemente, pode ser resolvido com dois CaU

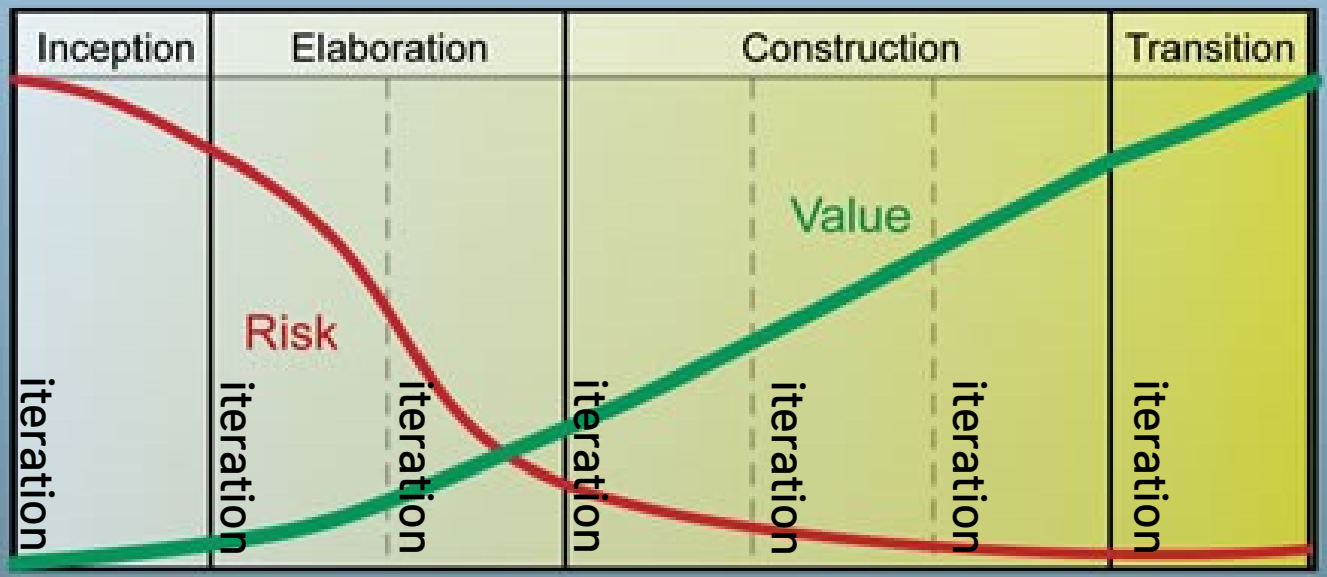
- O fluxo que cria
- O fluxo que consulta/atualiza



Desenvolvimento iterativo e incremental



Project Lifecycle



A B

Process: How to Work With Use Cases in Iterative Methods?

Use cases are central to the UP and many other iterative methods. The UP encourages **use-case driven development**. This implies:

- Functional requirements are primarily recorded in use cases (the Use-Case Model); other requirements techniques (such as functions lists) are secondary, if used at all.
- Use cases are an important part of iterative planning. The work of an iteration is—in part—defined by choosing some use case scenarios, or entire use cases. And use cases are a key input to estimation.
- **Use-case realizations** drive the design. That is, the team designs collaborating objects and subsystems in order to perform or realize the use cases.
- Use cases often influence the organization of user manuals.
- Functional or system testing corresponds to the scenarios of use cases.
- UI "wizards" or shortcuts may be created for the most common scenarios of important use cases to ease common tasks.

Prioritizar

Explicitar incrementos

Caraterística definidora: segmentar o âmbito do projeto em incrementos

“Refinar”

Os requisitos/casos de utilização podem ser detalhados à medida do necessário.

Use cases e os métodos ágeis → Use Cases 2.0

A granularidade dos casos de uso é algo "pesada" para a gestão do dia-a-dia

Proposta Use Cases 2.0

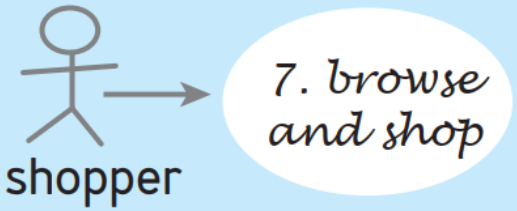
- "Fatias" de funcionalidade

Ponto de partida: use cases

- ...com a flexibilidade das *user stories/use case slices*

https://youtu.be/p5gDbf0je8k

Unidade mais conveniente: “fatia” de funcionalidade



priority: MUST
release: 1
size: very large
complexity: high

a use case and its properties captured on a sticky note

7.1 select and buy 1 product

flows: BF
test: 1 product, default payment, valid details

5

7.2 select and buy 100 products

flows: BF
test: 100 products, default payment, valid details

5

7.3 support systems unavailable

flows: BF, A9, A10, A1, A12
test: select product, provide information, disconnect each system in between

13

some slices from the use case captured on their own sticky notes