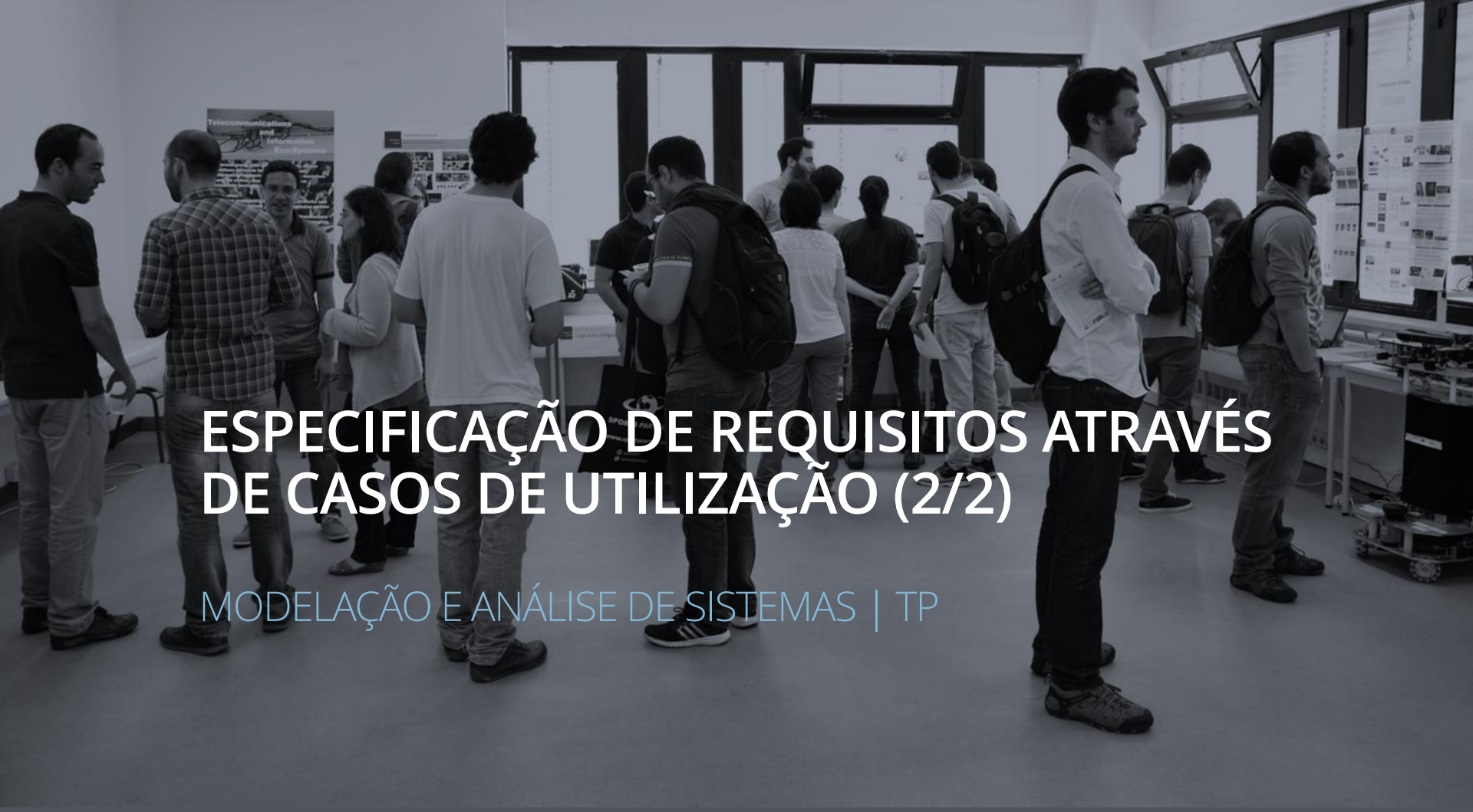


ESPECIFICAÇÃO DE REQUISITOS ATRAVÉS DE CASOS DE UTILIZAÇÃO (2/2)

MODELAÇÃO E ANÁLISE DE SISTEMAS | TP

ILÍDIO OLIVEIRA ico@ua.pt
v2018-03-01

universidade de aveiro
departamento de eletrónica, telecomunicações e informática
deti

“requisitos para uma casa...”

João, o pai

Requisitos

Sala de estar espaçosa e luminosa

Com lareira

Interpretando...

Espaçosa? 20, 100 m2?

Quantas pessoas?

Iluminação? Janelas? Varandas?

Maria, a mãe

Requisitos

Fácil acesso da cozinha para a sala
Não gosta de fornos elétricos
Gosta de banho de imersão e não de duche

Interpretando

Ligação entre sala e cozinha
Infraestrutura de gás
Uma case de banho com banheira

Luís, o filho

Requisitos

Gosta de jogos on-line, à noite
Uma cama com escorrega
Uma piscina grande para a sua tartaruga

Interpretação

Cobertura Wi-Fi
Confirmar: cama elevada, c/ escorrega
Piscina?!

Normas aplicáveis fazem parte da especificação

Pé-alto

Sistemas de energia

Capacidades exigidas, controlos de segurança,...

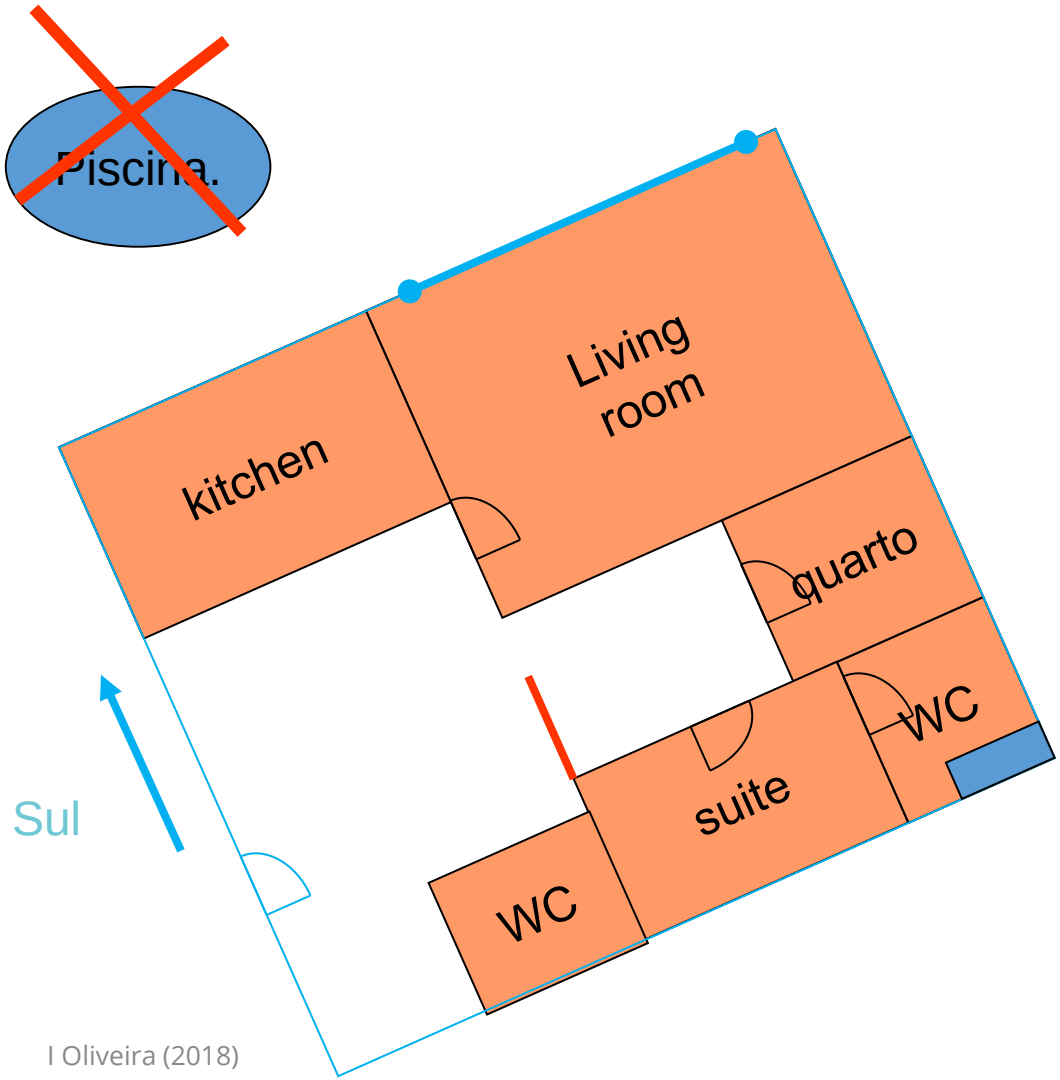
Espaço por pessoa

Condições para instalação de lareira

...

I Oliveira (2018)

Requisitos têm de ser selecionados



Requisitos essenciais *vs* facultativos

Capacidades *vs* Orçamento *vs* Tempo

Enquadramento regulamentar

Os atributos de qualidade são necessários para definir o produto

Um sistema tem funcionalidade e atributos de qualidade

Requisitos funcionais

Captam o comportamento pretendido do sistema

Expresso como serviços, funções ou tarefas que o sistema deve realizar

Pode ser captado nos CaU

Pode ser descrito com diagramas de comportamento: atividades, sequência

Requisitos não funcionais

Restrições globais num sistema de software

E.g.: robustez, portabilidade,...

Também designados como atributos de qualidade

Por regra, não afetam apenas um módulo/CaU

TABLE 14-1 Some software quality attributes

External quality	Brief description
Availability	The extent to which the system's services are available when and where they are needed
Installability	How easy it is to correctly install, uninstall, and reinstall the application
Integrity	The extent to which the system protects against data inaccuracy and loss
Interoperability	How easily the system can interconnect and exchange data with other systems or components
Performance	How quickly and predictably the system responds to user inputs or other events
Reliability	How long the system runs before experiencing a failure
Robustness	How well the system responds to unexpected operating conditions
Safety	How well the system protects against injury or damage
Security	How well the system protects against unauthorized access to the application and its data
Usability	How easy it is for people to learn, remember, and use the system
Internal quality	**Brief description**
Efficiency	How efficiently the system uses computer resources
Modifiability	How easy it is to maintain, change, enhance, and restructure the system
Portability	How easily the system can be made to work in other operating environments
Reusability	To what extent components can be used in other systems
Scalability	How easily the system can grow to handle more users, transactions, servers, or other extensions
Verifiability	How readily developers and testers can confirm that the software was implemented correctly

Components of **FURPS+**

F unctionality	Feature set capabilities, security, generality	
U sability	Human factors aesthetics, consistency, documentation	
R eliability	Frequency/severity of failure, recoverability, predictability, accuracy, MTBF	
P erformance	Speed efficiency, resource usage, throughput, response time	
S upportability	Testability Adaptability Compatibility Serviceability Localizability	Extensibility Maintainability Configurability Installability Robustness

Grady, 1992

Requirements
STRQ1: Want to be able to transfer funds from other accounts (not necessarily held with this firm) to a trading account.
STRQ2: State and federal regulations require monthly reports of account activity. Refer to specification RUFS-1234 for details of the information required.
STRQ3: The system should allow the use of any browser.
STRQ4: Customers want to manage their retirement funds.
STRQ5: Must be able to upgrade the system without taking it offline.
STRQ6: The system should allow traders to trade in multiple markets across the world.
STRQ7: Must be able to provide convenient answers to customer's most common questions.
STRQ8: The system must provide a secure environment that prohibits fraudulent access.
STRQ9: Need a way to train customers in the use of the system quickly and conveniently.
STRQ10: The system must operate on hardware that falls under the company's current maintenance contracts.
STRQ11: Need to be able to maintain the system with our current IT hardware and skills. Refer to enterprise architecture document EA-1234 for details.
STRQ12: Need account activity statements for tax reporting.
STRQ13: The system must provide all the basic capabilities of a normal stock broking firm.
STRQ14: Need to be able to perform research on any given stock.
STRQ15: The system must allow traders to obtain up-to-date news and alerts on nominated stock.
STRQ16: The system must provide current and historical information on Trading Acounts. Such as number of shares held, current price, total Trading Account value
STRQ17: The system shall provide the following types of trades: Market Trades (buy and sell), Limit Trades (buy and sell), and transfers between mutual funds.

S. M. A. R. T.

Step 5: Specify well-structured quality requirements

Simplistic quality requirements such as "The system shall be user-friendly" or "The system shall be available 24x7" aren't useful. The former is far too subjective and vague; the latter is rarely realistic or necessary. Neither is measurable. Such requirements provide little guidance to developers. So the final step is to craft specific and verifiable requirements from the information that was elicited regarding each quality attribute. When writing quality requirements, keep in mind the useful SMART mnemonic—make them *Specific*, *Measurable*, *Attainable*, *Relevant*, and *Time-sensitive*.

"The system shall be 100% reliable and 100% available".

"The system shall have a minimum response to a query of 1 second irrespective of system load".

AVL-1. The system shall be at least 95 percent available on weekdays between 6:00 A.M. and midnight Eastern Time, and at least 99 percent available on weekdays between 3:00 P.M. and 5:00 P.M. Eastern Time.

IOP-1. The Chemical Tracking System shall be able to import any valid chemical structure from the ChemDraw (version 13.0 or earlier) and MarvinSketch (version 5.0 or earlier) tools.

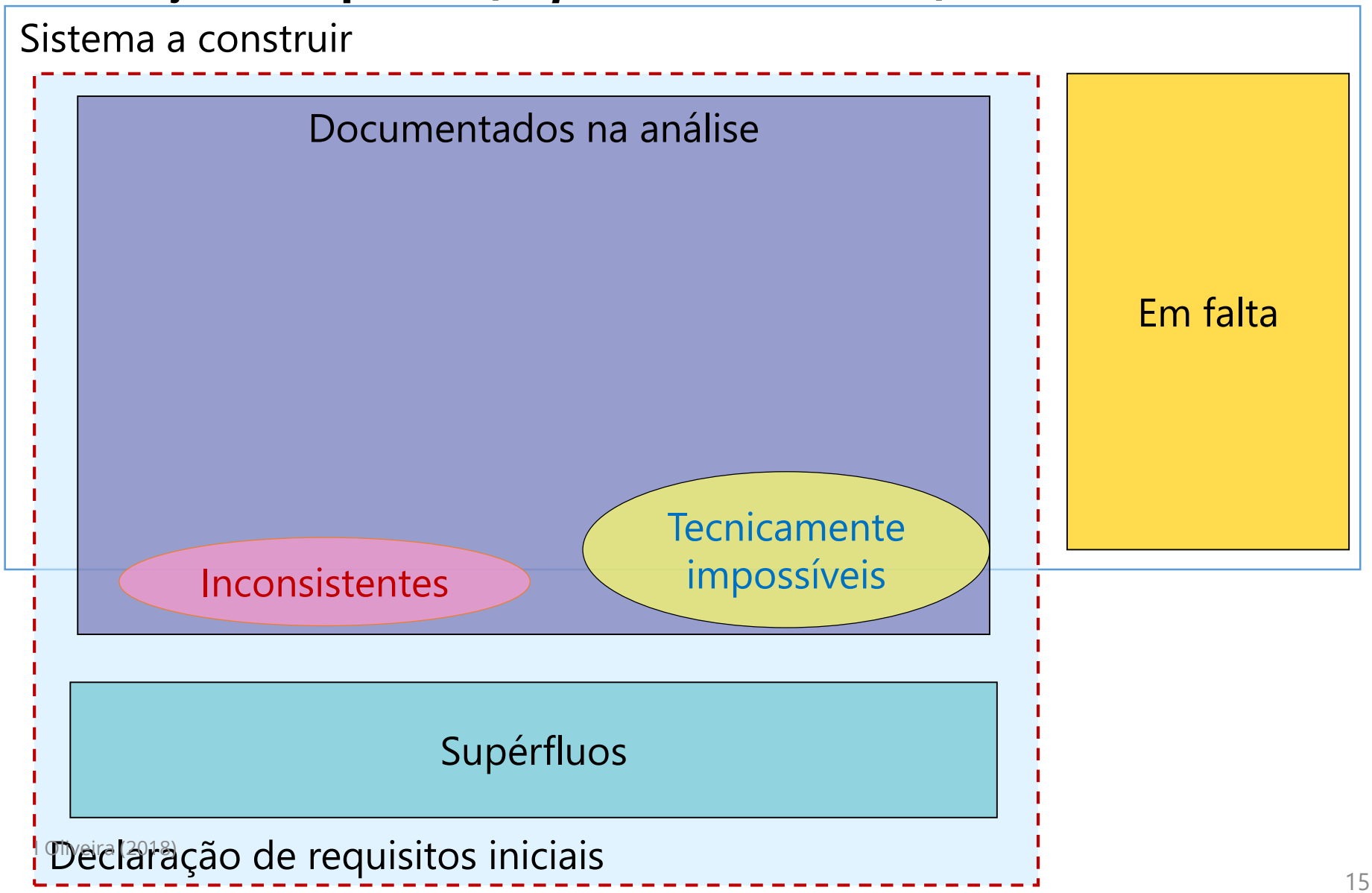
PER-1. Authorization of an ATM withdrawal request shall take no more than 2.0 seconds.

PER-2. The anti-lock braking system speed sensors shall report wheel speeds every 2 milliseconds with a variation not to exceed 0.1 millisecond.

PER-3. Webpages shall fully download in an average of 3 seconds or less over a 30 megabits/second Internet connection.

PER-4. At least 98 percent of the time, the trading system shall update the transaction status display within 1 second after the completion of each trade.

Dedução *de* requisitos (*Requirements elicitation*)



I. Oliveira (2018)

Requisitos são muitas vezes elencados em listas “o sistema deve…”

#	Requisito
RF.1	O sistema deve permitir a um profissional criar um novo pedido de adesão , em auto-serviço, na web.
RF.2	O sistema deve enviar credenciais temporárias para os pedidos de adesão e enviá-las, por email, aos solicitantes.
RF.3	O sistema deve permitir a pesquisa de cheques-dentista (emitidos) por número de utente do SNS.
RNF.1	As pesquisas de cheques-dentista têm de retornar resultados em <5 segundos ou um evento de tempo expirado.
…	

Sistemas de sw especificados à maneira de contratos

SRS – Software Requirements Specification

A apresentação de requisitos Segundo um modelo internacional:

Norma ISO/IEC/IEEE 29148:2011: Systems and software engineering -- Life cycle processes --Requirements engineering

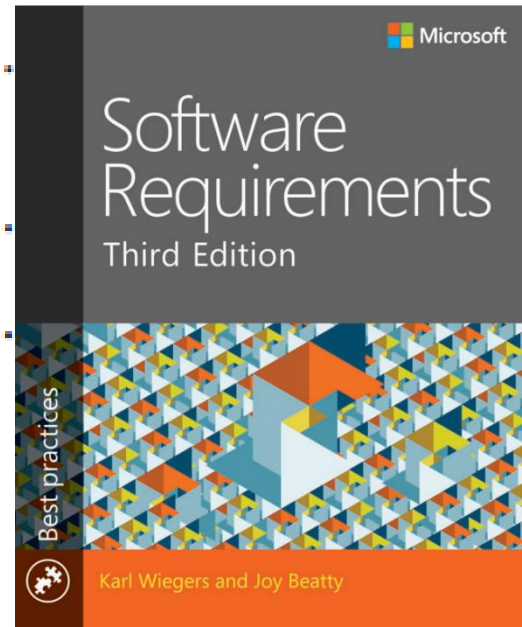
Outros modelos: Wiegers'

A software requirements specification template . . . . . .

1. Introduction . . . . . .
2. Overall description. . . . . .
3. System features . . . . . .
4. Data requirements. . . . . .
5. External interface requirements . . . . . .
6. Quality attributes. . . . . .
7. Internationalization and localization requirements . .
8. [Other requirements]. . . . . .

Appendix A: Glossary . . . . . .

Appendix B: Analysis models . . . . . .

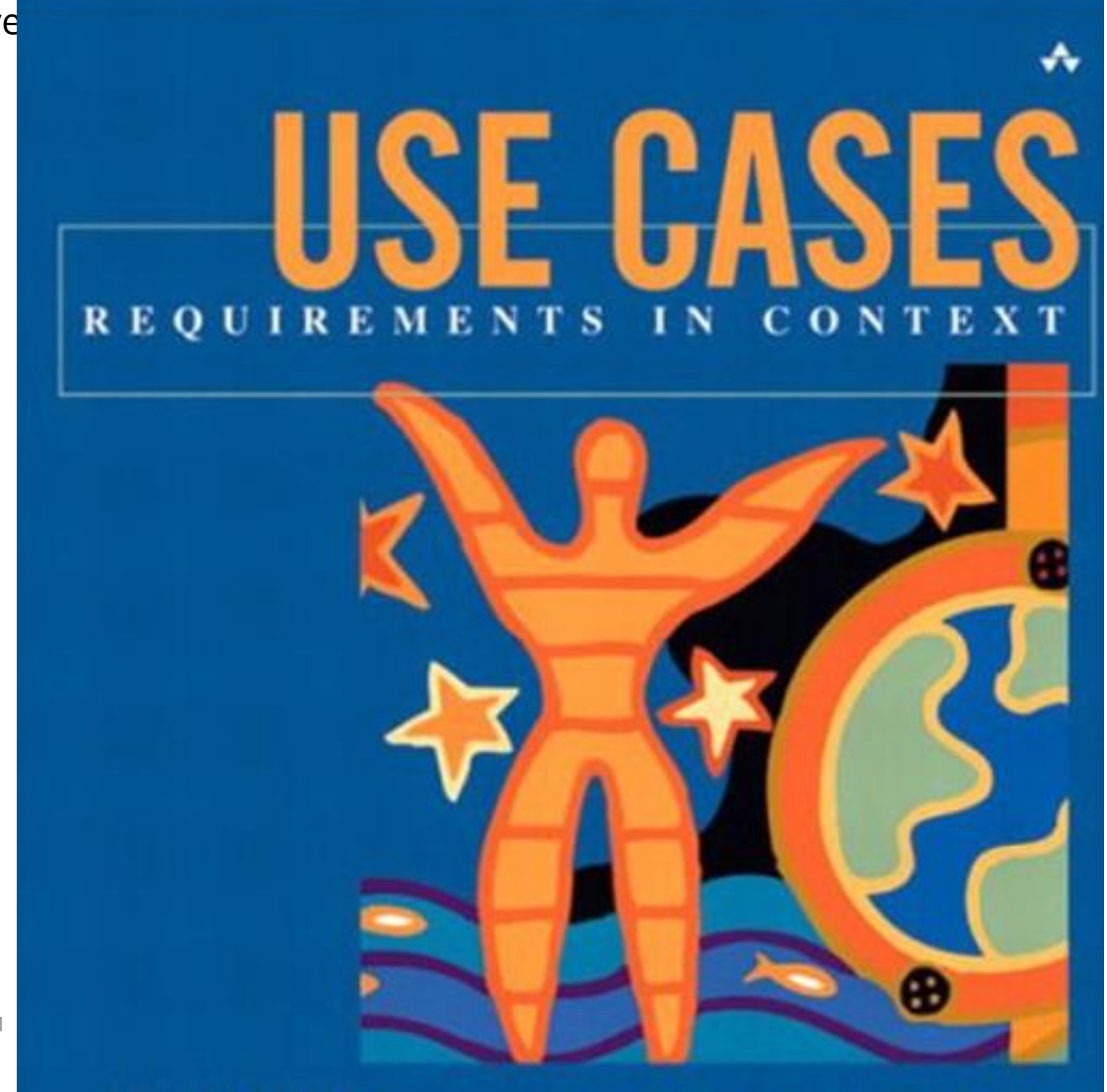


Qual o nossa estratégia para documentar os requisitos?

CaU contam histórias que mostram os requisitos funcionais em contexto

CaU:	Consultar informação clínica (referenciação)
Propósito:	O Médico Dentista acede ao sistema para consulta infromação clínica inserida pelo médico assistente.
Sequência típica:	Inicia-se quando o Médico Dentista recebe um Utente portador de CD para consulta. O MD acede à opção de pesquisa na sua página de entrada. O sistema apresenta o formulário de pesquisa. O MD insere o número de utente do SNS e confirma. O sistema pesquisa os cheques-dentista existentes para aquele utente e apresenta uma listagem ordenada do mais recente para o mais antigo. O MD seleciona uma entrada na lista para abrir a informação de detalhe. O sistema apresenta para esse CD o cabeçalho com a identificação do cheque e utente, e uma seção com a informação clínica disponível.
Sequências alternativas:	...

l Olive



Narrativas dos CaU ligam ao Modelo do Domínio

The user selects a title *and adds it to his* list of books to be saved for later. *The system displays a page with the updated list and also shows a* list of titles in the user's cart, ready for checkout.

The user selects a ***Book*** *and adds it to his* ***Wish List****. The system displays a page with the updated list and also displays the user's* ***Shopping Cart****.*

I Oliveira (2018)

CaU Essencias vs Concretos

Estilo essencial: manter as referências ao interface com o utilizador de parte e focar na intenção do ator.

1. Administrator identifies self.
2. System authenticates identity.
3. ...

vs.

1. Adminstrator enters ID and password in dialog box (see Picture 3).
2. System authenticates Administrator.
3. System displays the "edit users" window (see Picture 4).
4. ...

E os requisitos não funcionais?

Ver modelo proposto

Secção Requisitos especiais

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>*

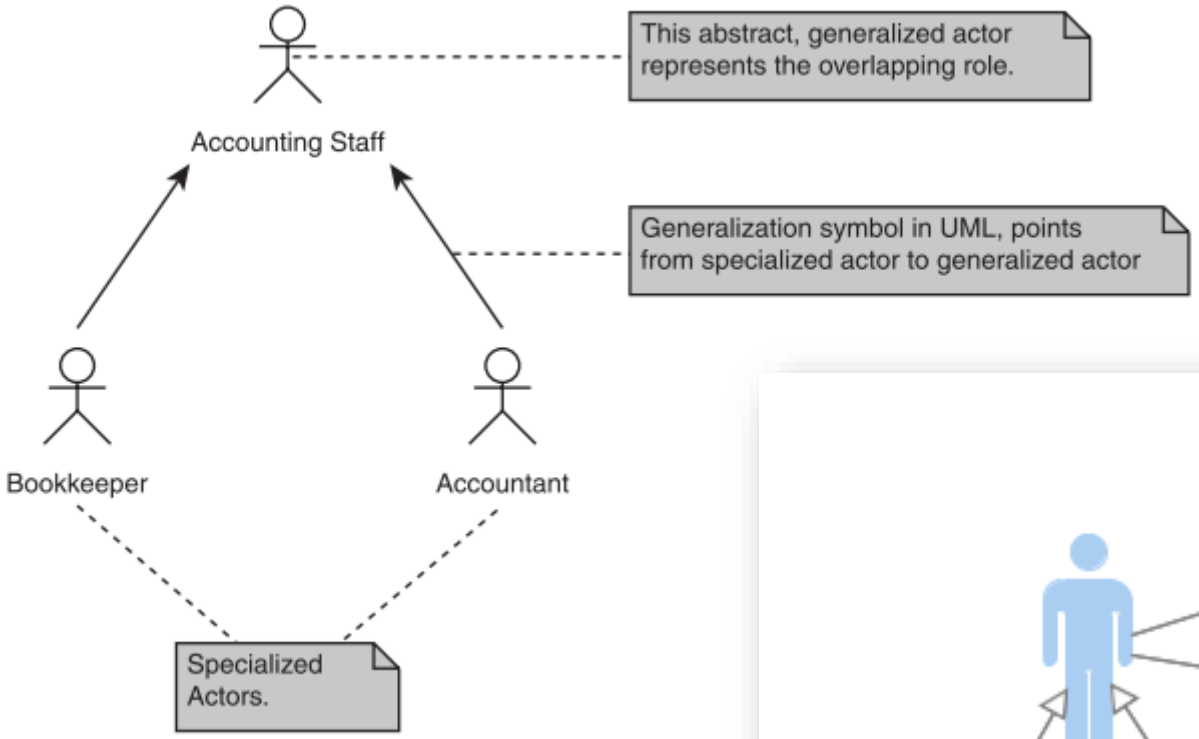
<step altered> <condition>: <action or sub use case>

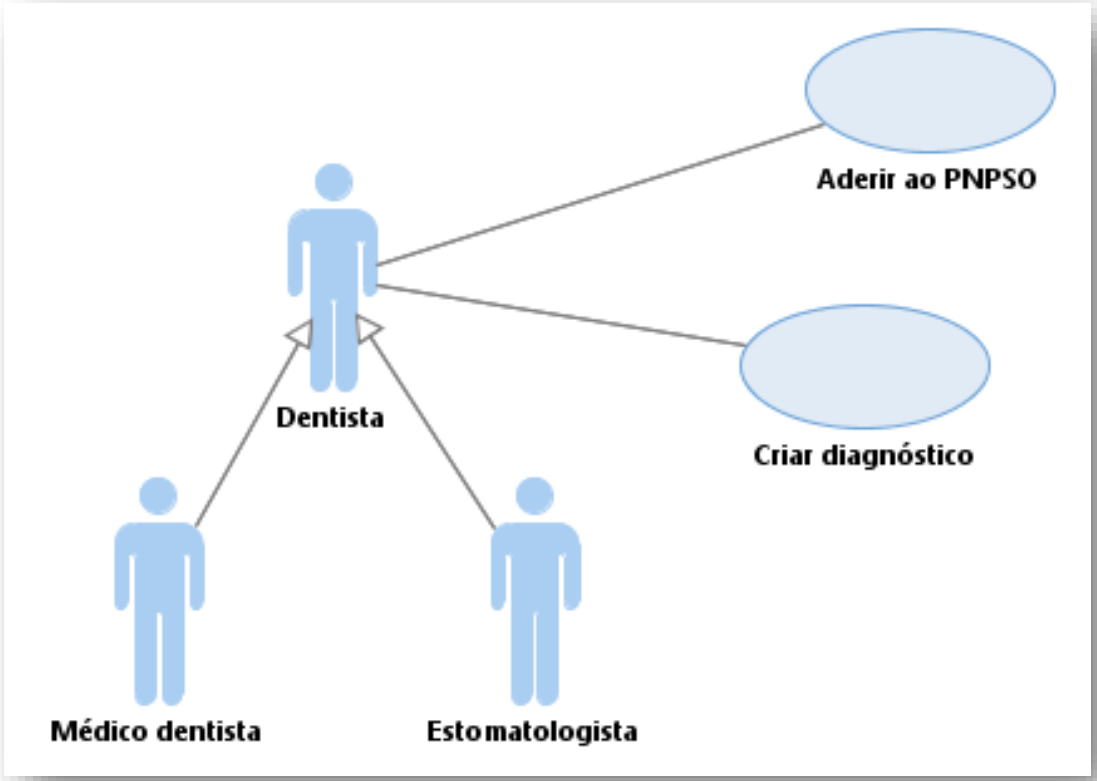
<step altered> <condition>: <action or sub use case>

Technology & Data Variations List:

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

Pode haver especialização entre atores (i.e., papéis)





Atores primários e secundários

Ator primário

Solicita o sistema para resolver problemas/realizar objetivos

Os CaU são iniciados por um Ator primário

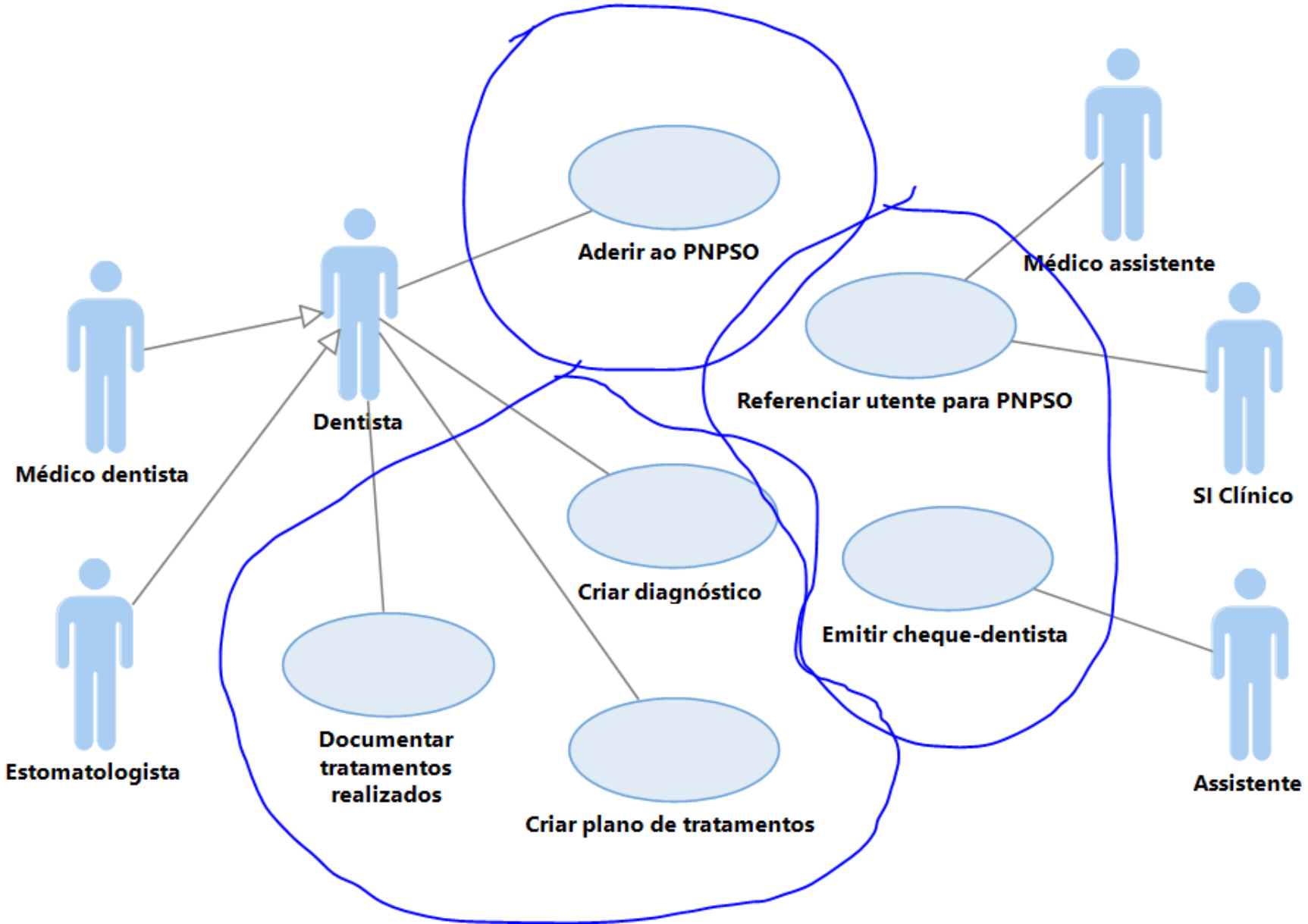
Ator secundário

Fornece serviços ou informação para algum cenário do CaU

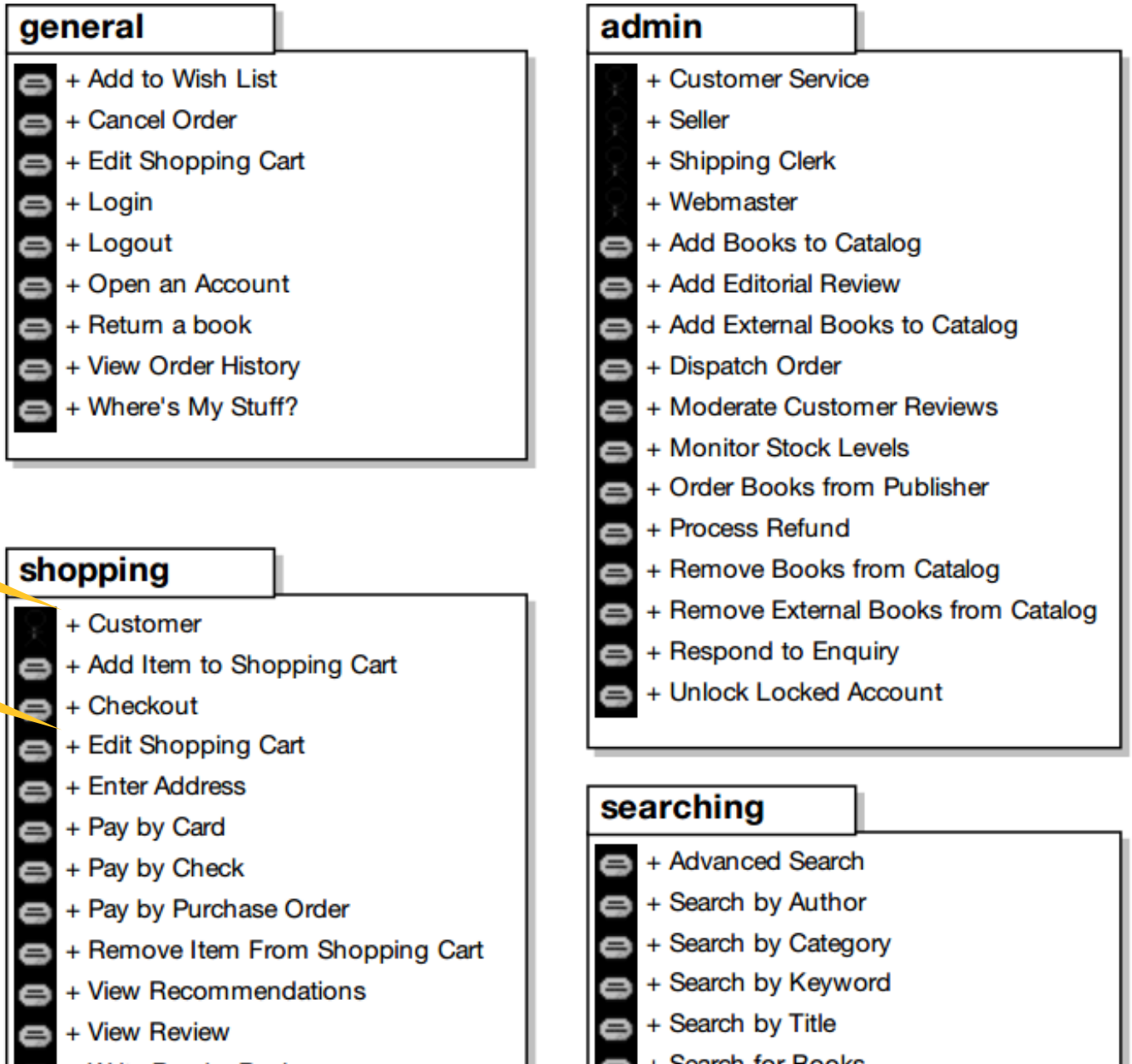
Podem ser sistemas externos ou papéis de pessoas, que não são utilizadores

I Oliveira (2018)

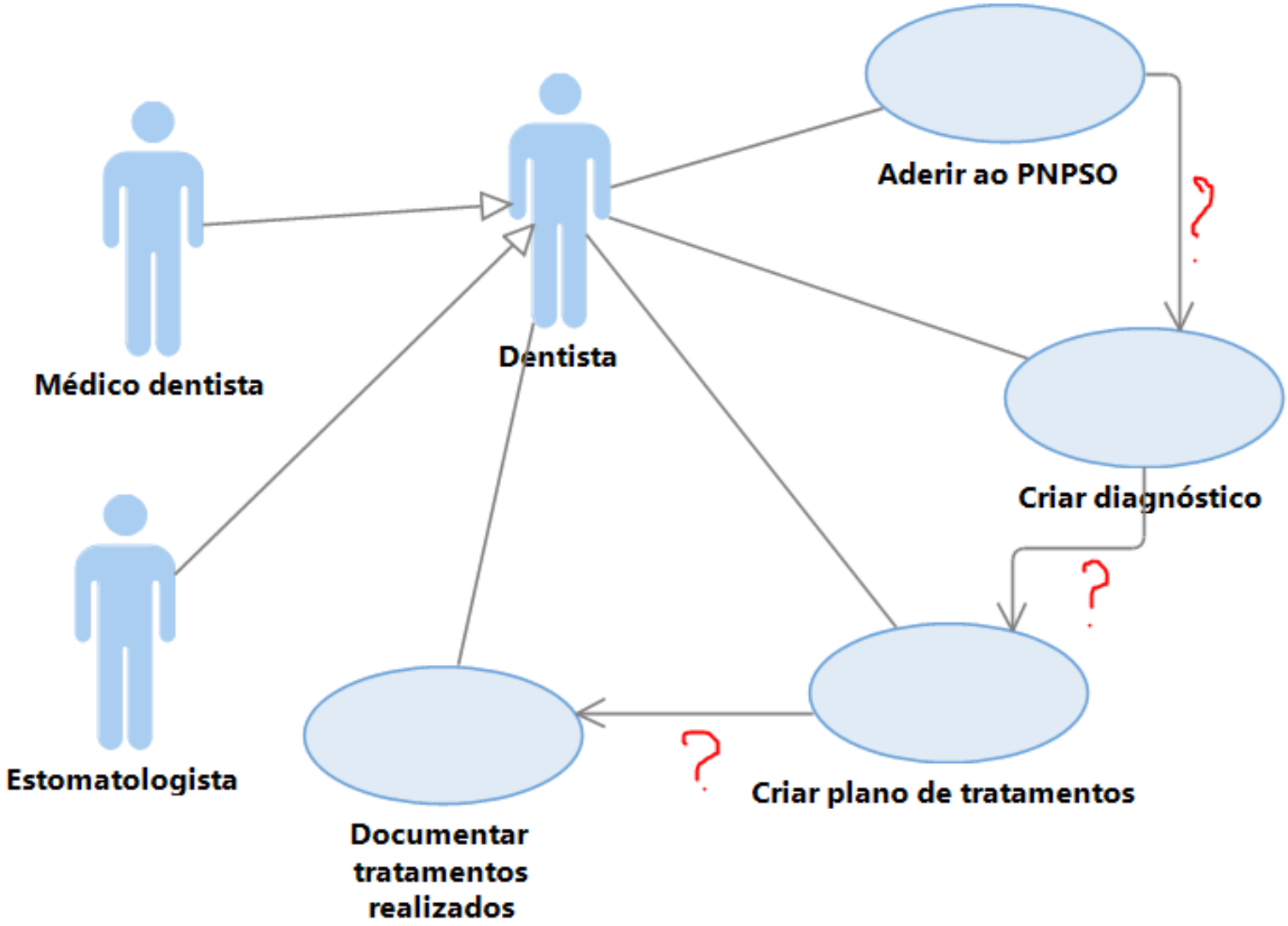
Podemos agrupar os casos de utilização em temas



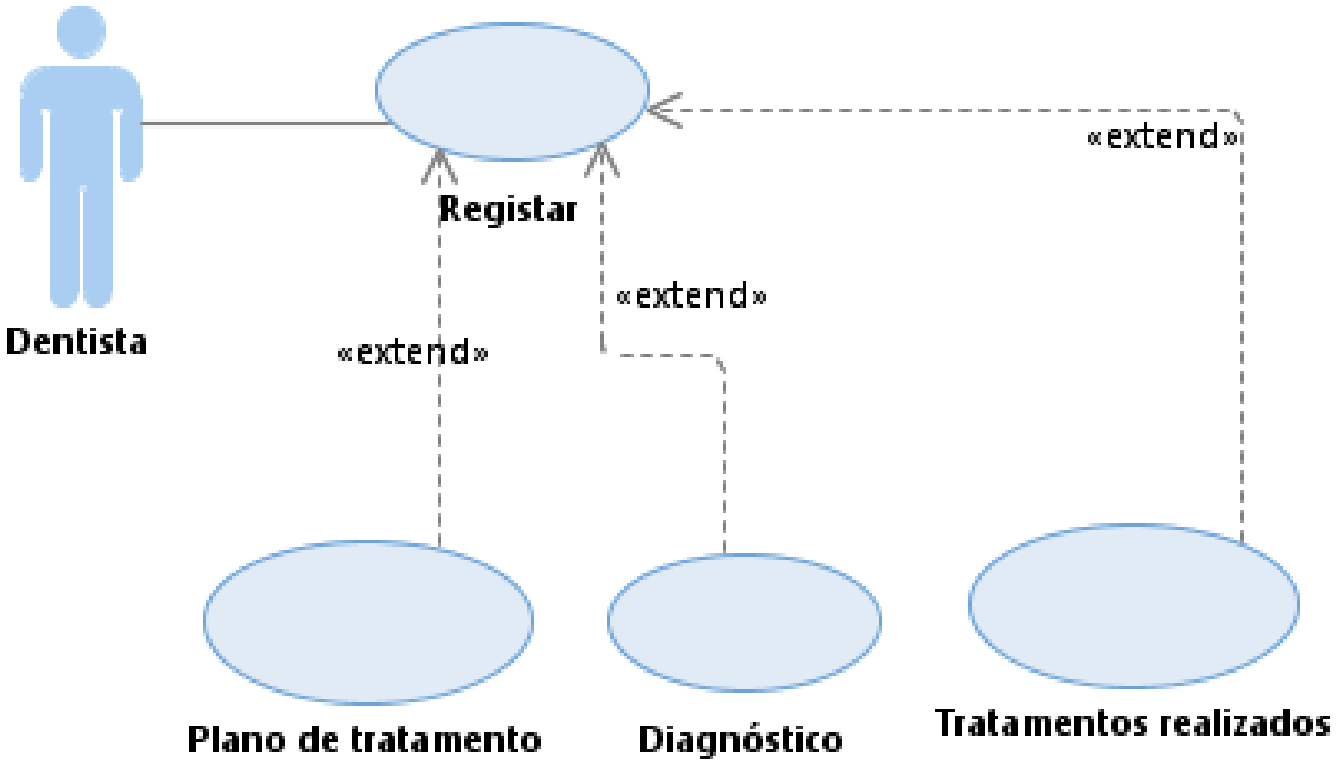
Os casos de


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
Actors
Use cases

CaU não mostram *workflow*



Decomposição/Agregação funcional

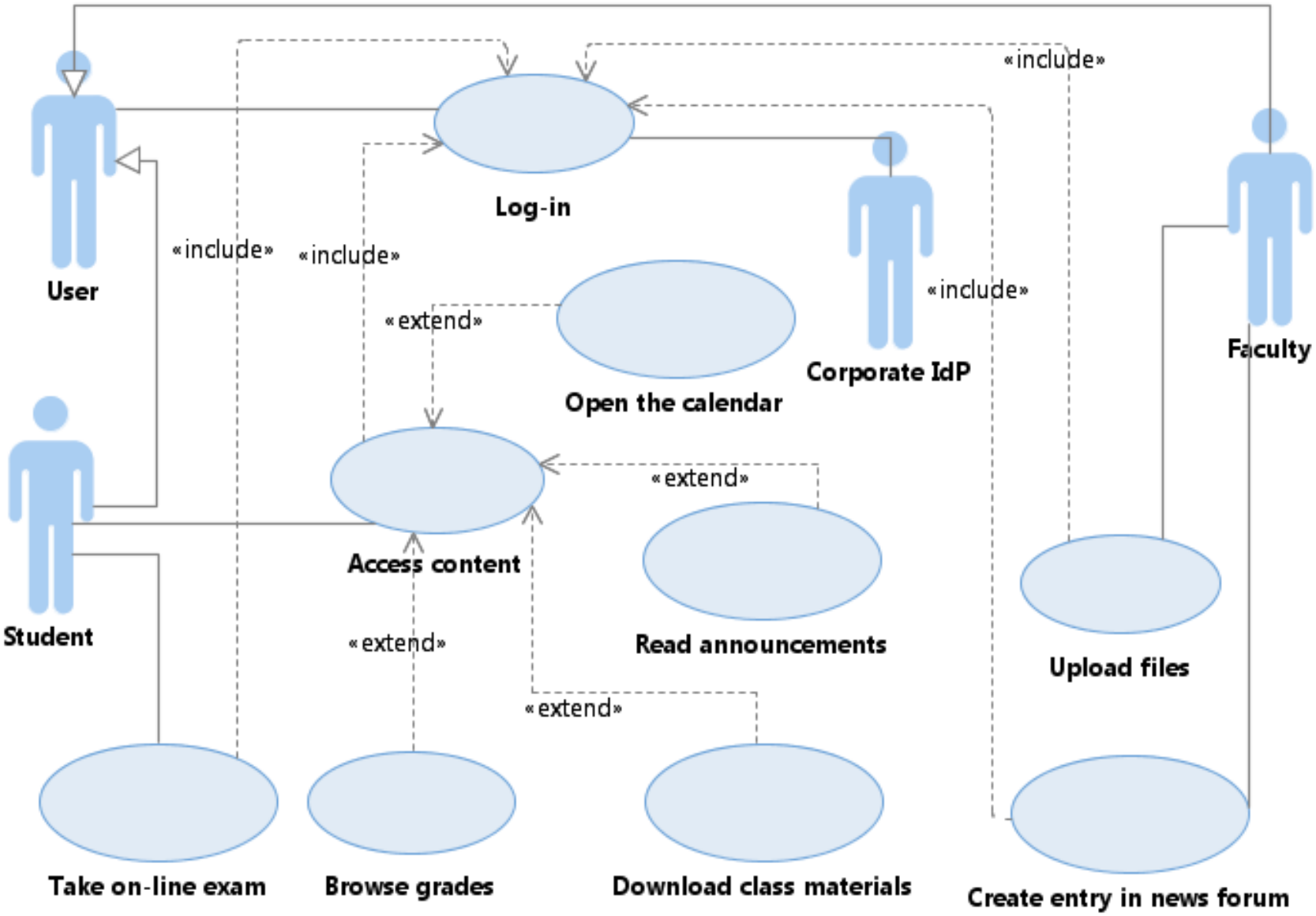


I Oliveira (2018)

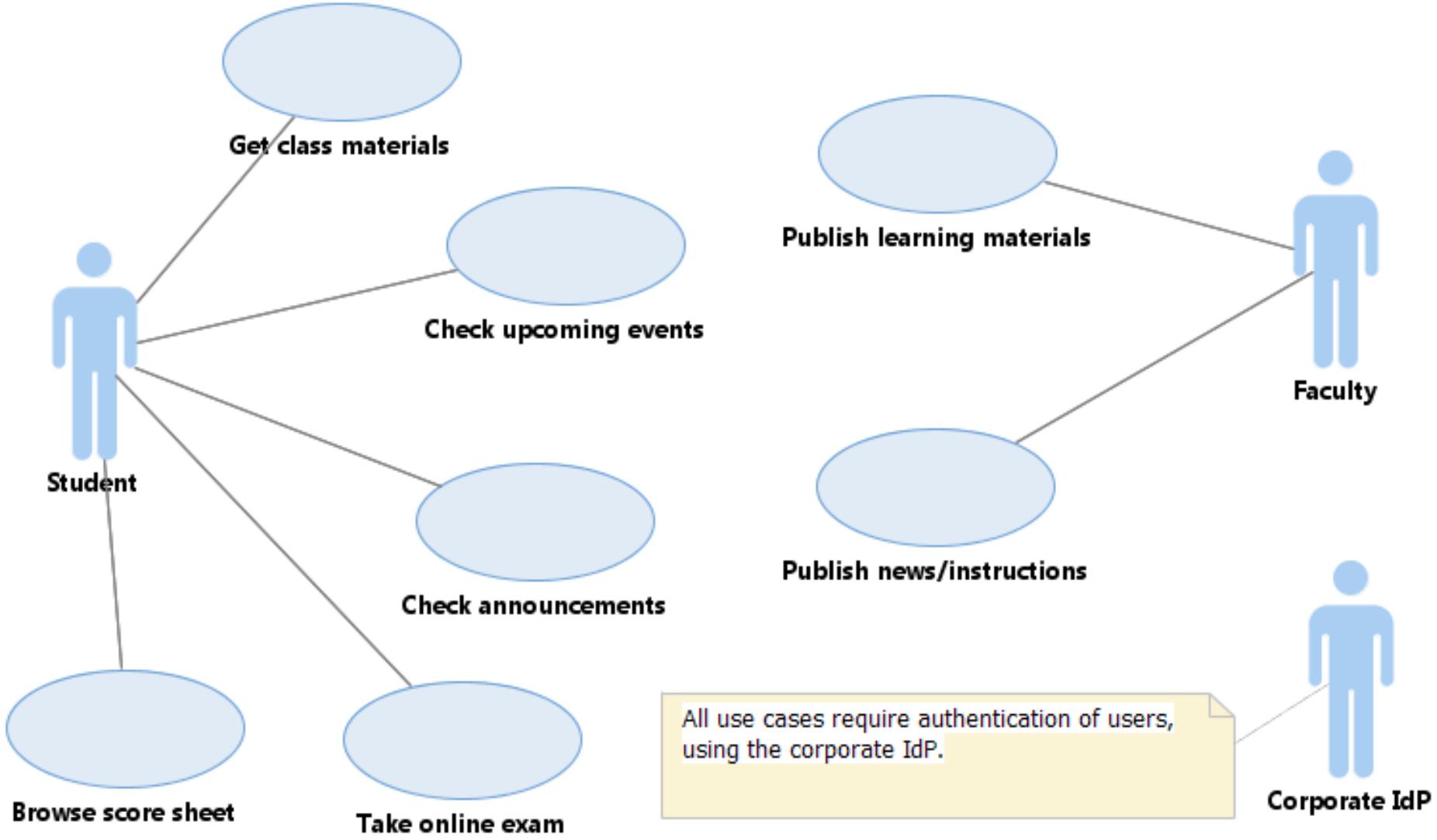
Login e Logout são casos de utilização?

I Oliveira (2018)

Casos de utilizacão Moodle – Opcão 1

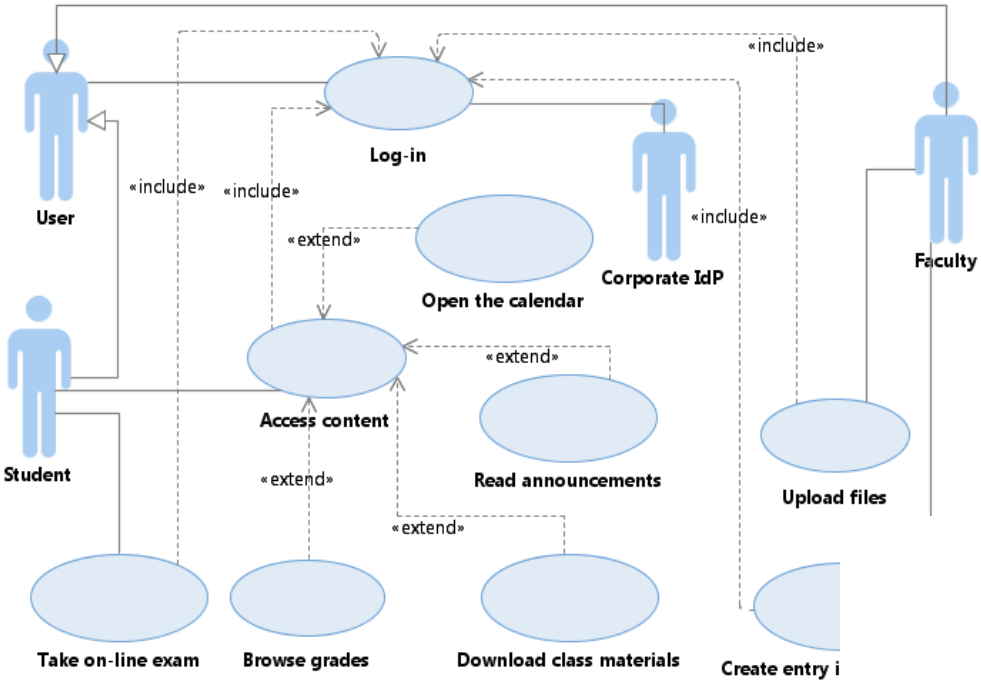


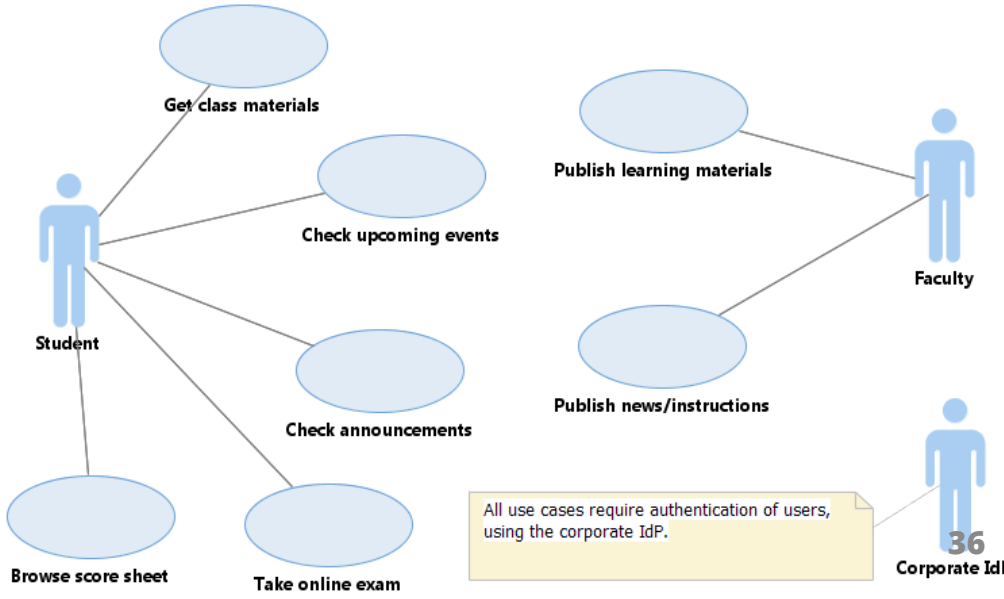
Casos de utilizacão Moodle – Opção 2



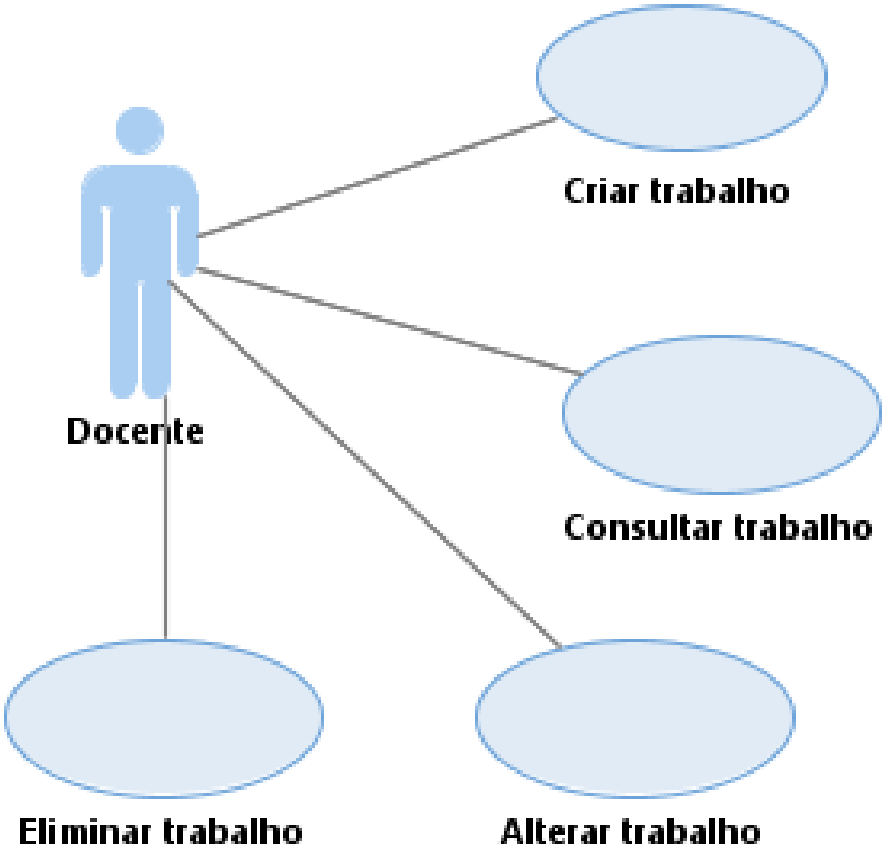
I Oliveira (2018)

Correção? Clareza? Eficácia?





Como lidar com o CRUD?



I Oliveira (2018)

Como lidar com o CRUD?

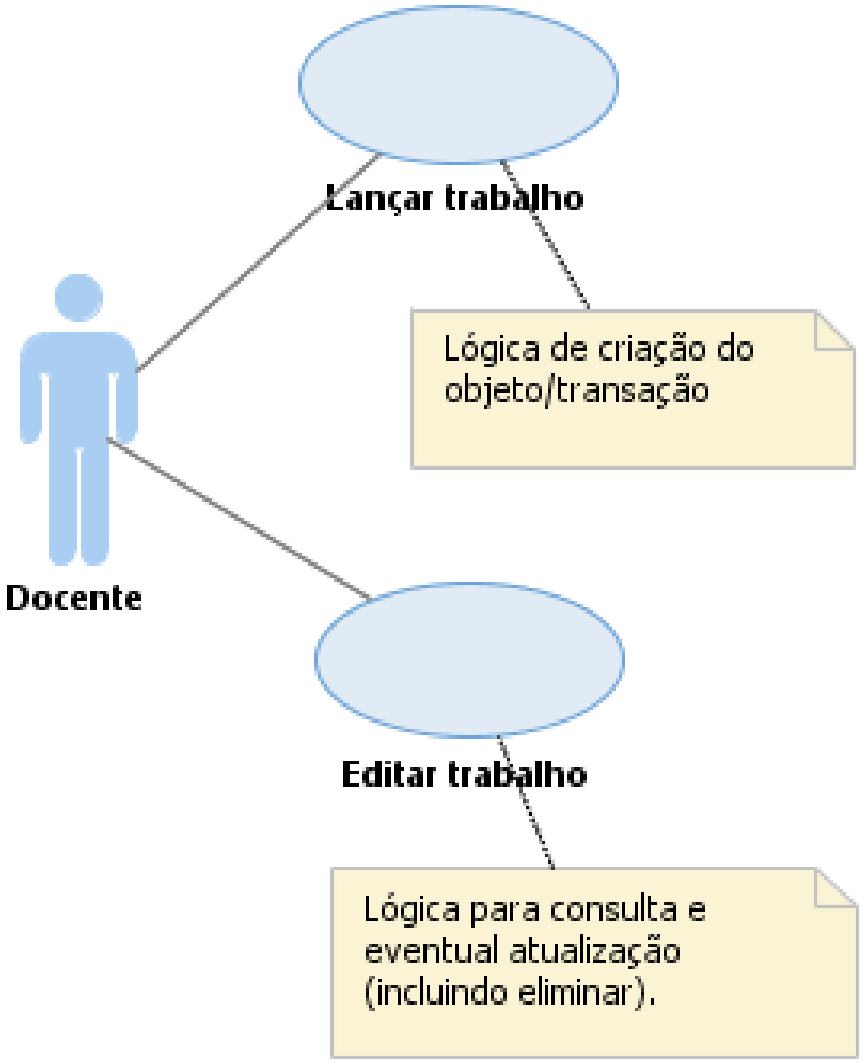
Depende do problema

Cancelar Cheque-dentista vs
Cancelar trabalho (Moodle)

Frequentemente, pode ser resolvido com dois CaU

O fluxo que cria

O fluxo que consulta/atualiza

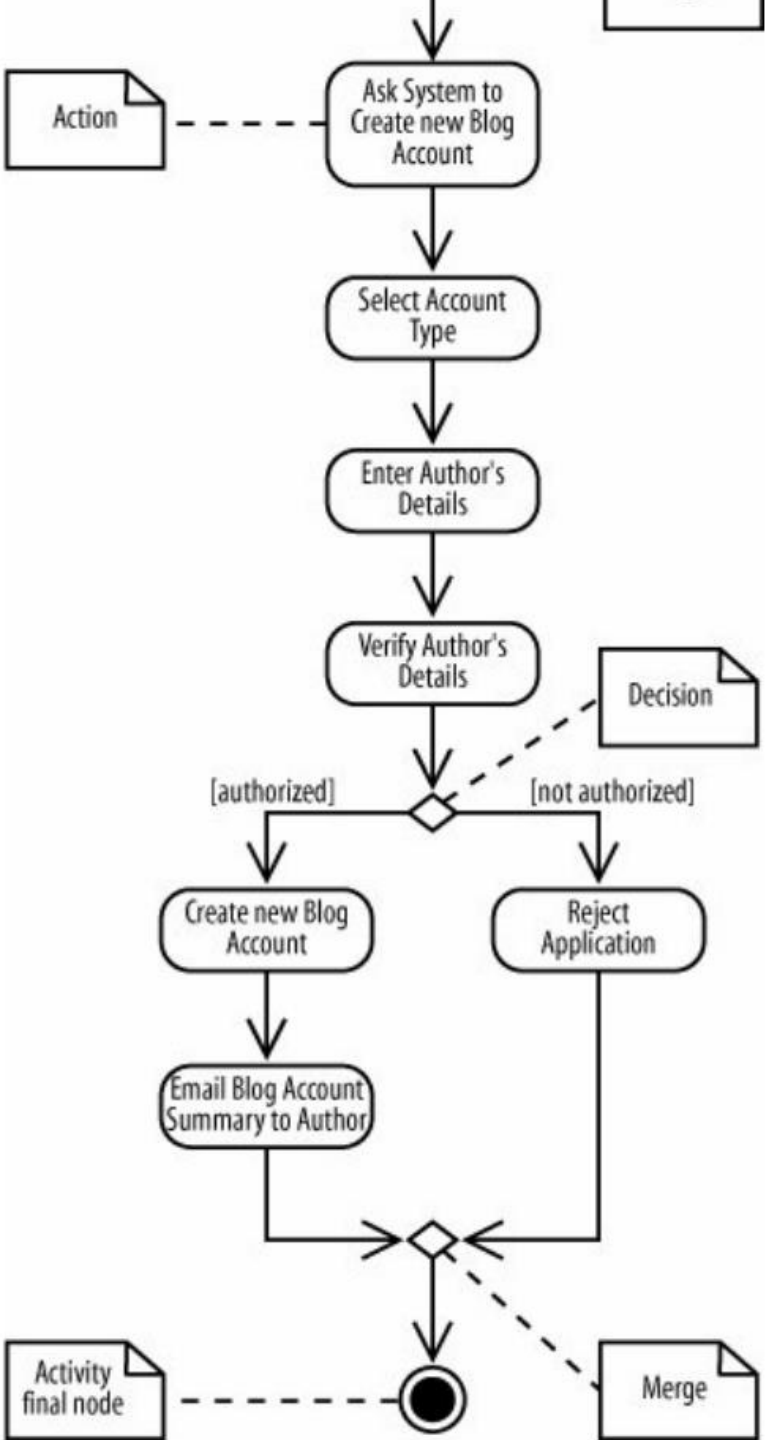


Explicar os fluxos com DA

Create a new Blog Account use case des

Step	Action
1	The Administrator asks the system to create a
2	The Administrator selects an account type.
3	The Administrator enters the author's details.
4	The author's details are verified using the Aut Database.
5	The new blog account is created.
6	A summary of the new blog account's details a author.

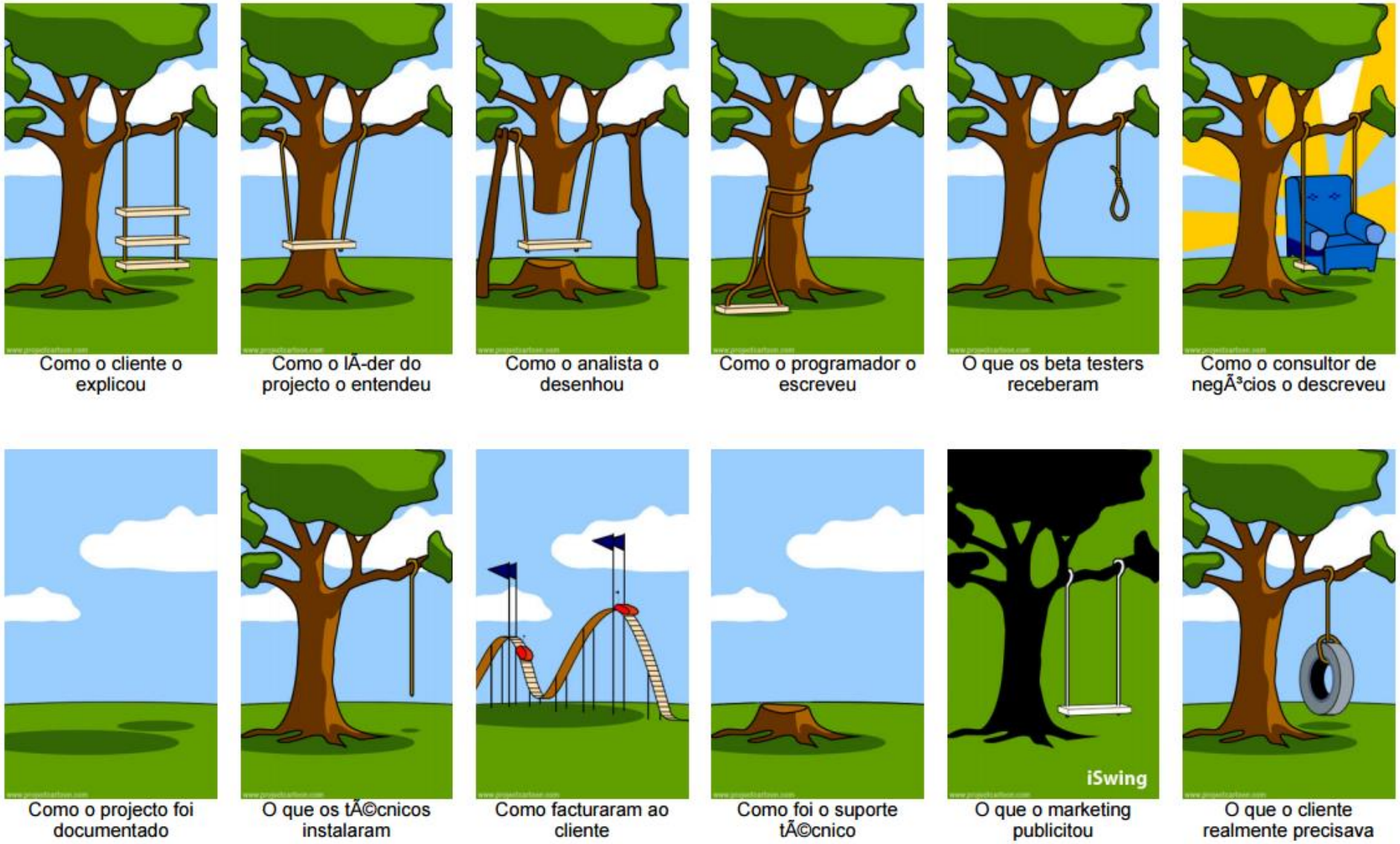
Step	Branching Action
4.1	The Author Credentials Database does not ver



Explicar os fluxos com D. Sequência

À suivre…

I Oliveira (2018)



http://projectcartoon.com

Referências

[PRE'10] Pressman, R. S. (2010). Software Engineering: a practitioners approach (seventh ed). McGraw Hill.

→ Chap. 5

[DEN'15] Dennis, A., Wixom, B. H., & Tegarden, D. (2015). Systems analysis and design: An object-oriented approach with UML. John Wiley & Sons.

→ Chap. 3

[LAR'12] Larman, C. (2012). Applying UML and Patterns: An Introduction to Object Oriented Analysis and Design and Interative Development. Pearson Education.

→ chap. 5.