

41951- ANÁLISE DE SISTEMAS

# Visual Modelling with UML

Ilídio Oliveira | v2022/03/08

# Learning objectives for this lecture

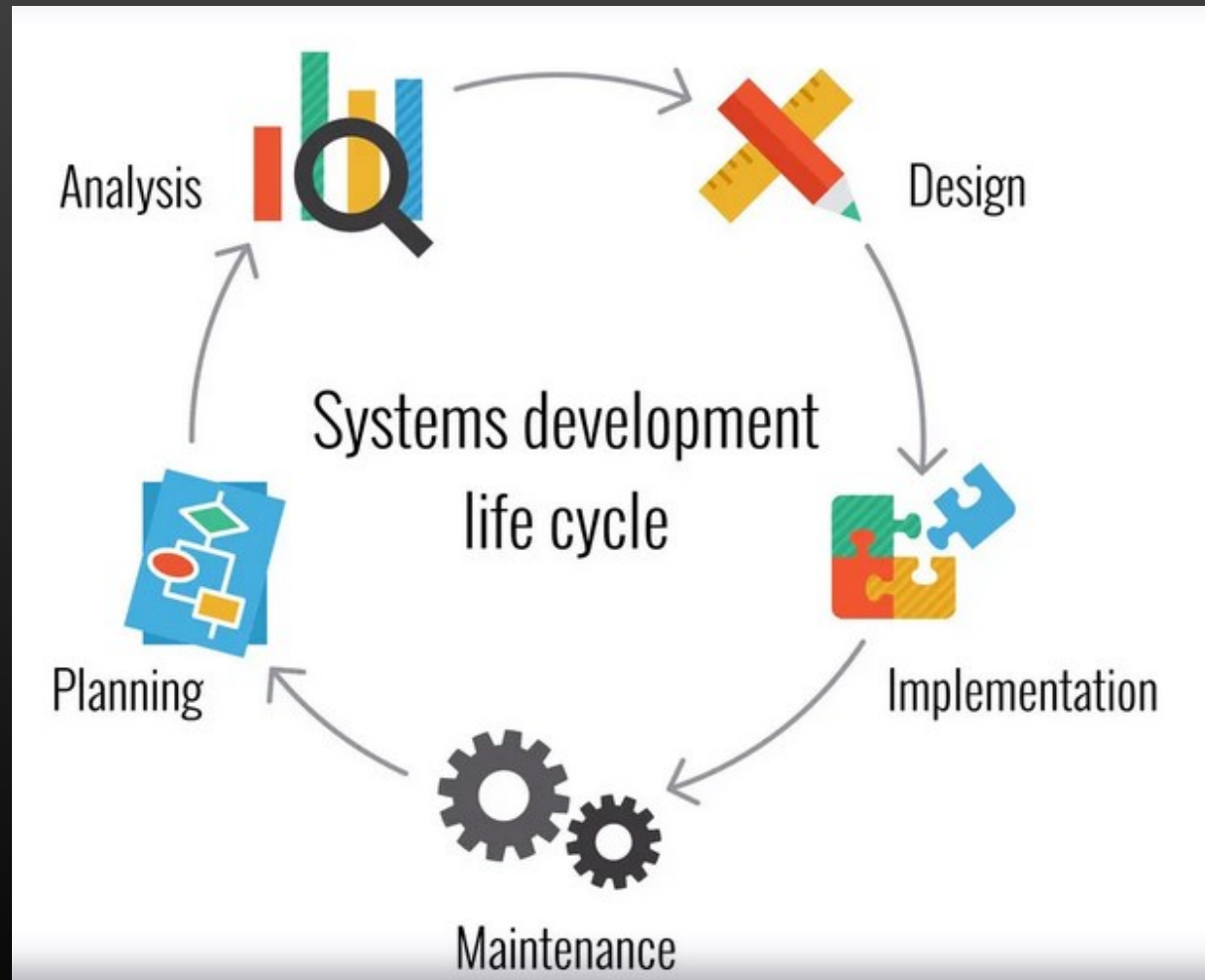
**Justify the use of models in systems engineering**

**Enumerate advantages of visual models**

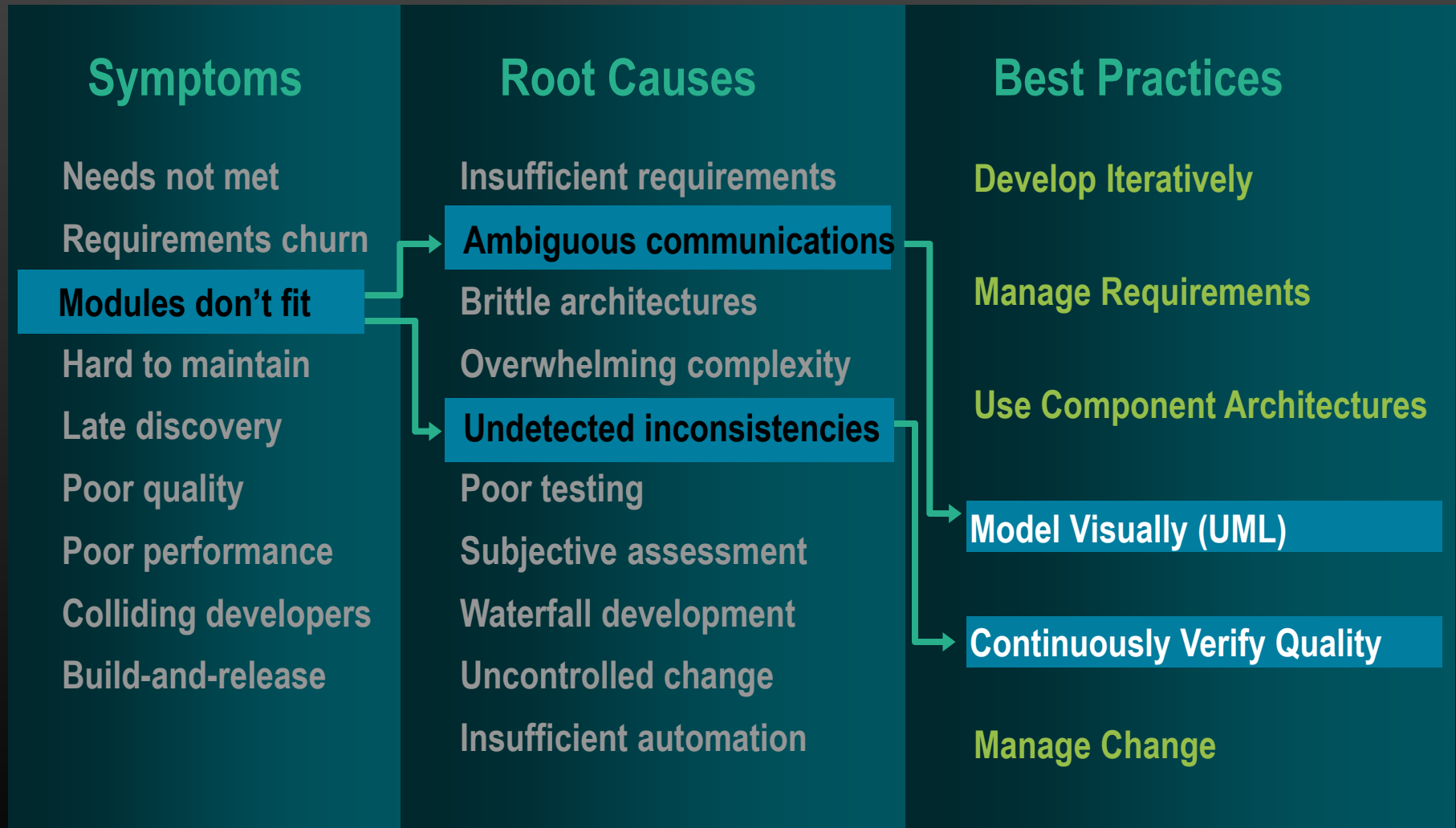
**Explain the organization of the UML**

**Identify the main diagrams in UML and their modeling viewpoint**

# Systems development lifecycle (SDLC)



# Problems and solutions in the SDLC (a Rational Unified Process perspective)



# Modeling

UML as a visual specification language



# Usamos modelos visuais para captar partes do mundo/realidade

D Trumpet Version

**Allegro Assai**  
from  
**Brandenburg Concerto #2**  
J. S. Bach  
arranged by Mark Adler

Allegro assai *tr*  
*mf*

Trumpet

Allegro assai  
*mf*

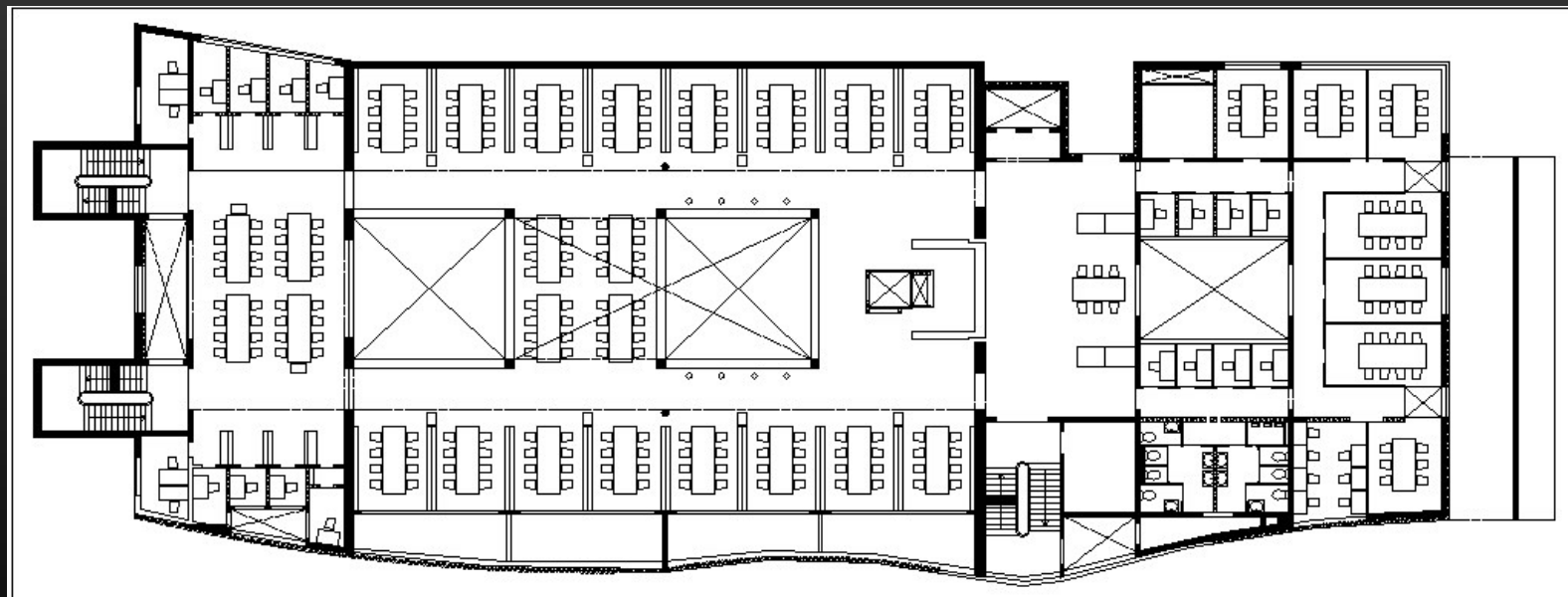
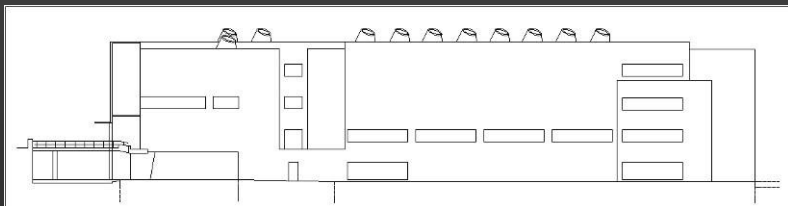
Organ



- Uma linguagem comum (escrever, ler)
- Especificações visuais são mais inteligíveis
- Compor: aplicar talento e disciplinas técnicas
- Orquestra: a prova que os modelos funcionam!



# Um modelo é uma simplificação da realidade



# Os modelos ajudam a gerir a complexidade

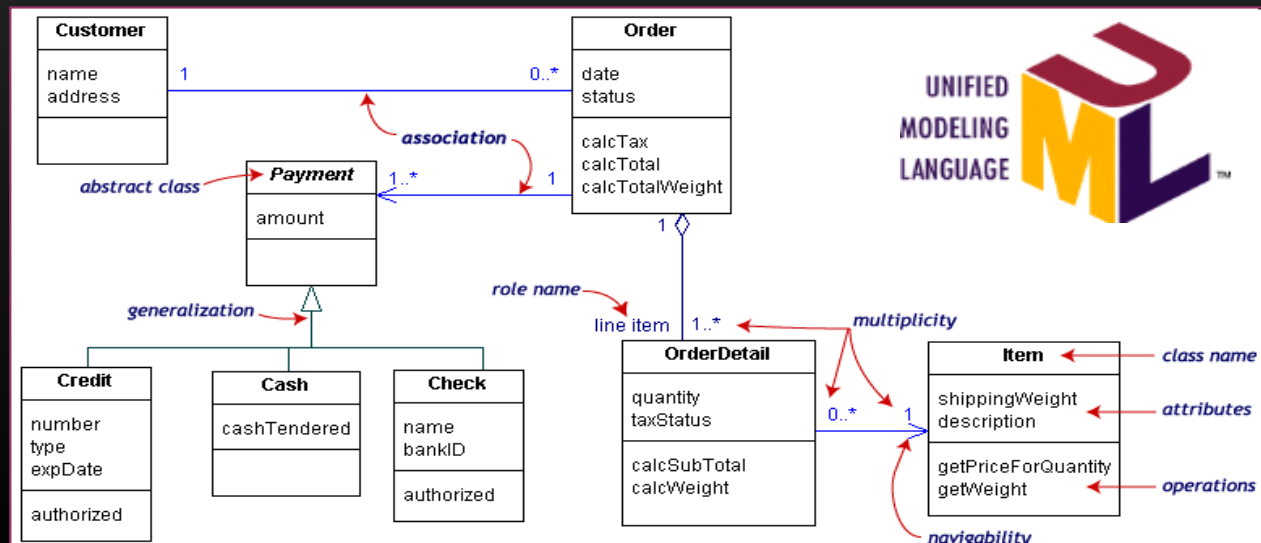
## 4 razões para usar modelos (G. Booch):

Ajudar a **visualizar** um sistema (*high-level*)

Especificar/documentar a **estrutura e o comportamento** do sistema (antes de implementar)

Serve como **referência** para orientar construção ("planta")

**Documentar as decisões** (de desenho) que foram feitas





# Modelação visual no desenvolvimento

## UML 2: Unified Modeling Language

Linguagem de modelação normalizada

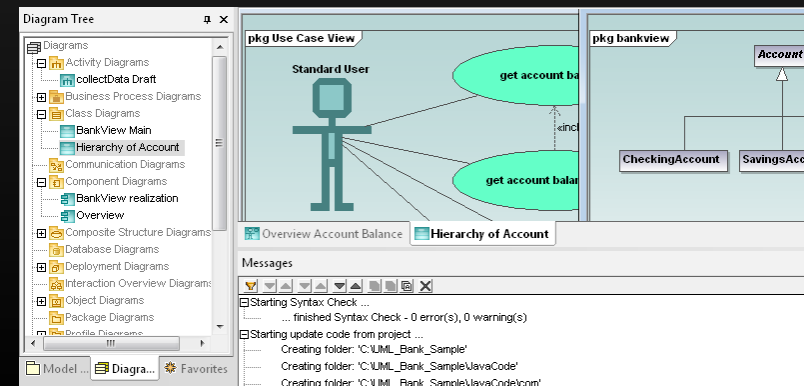
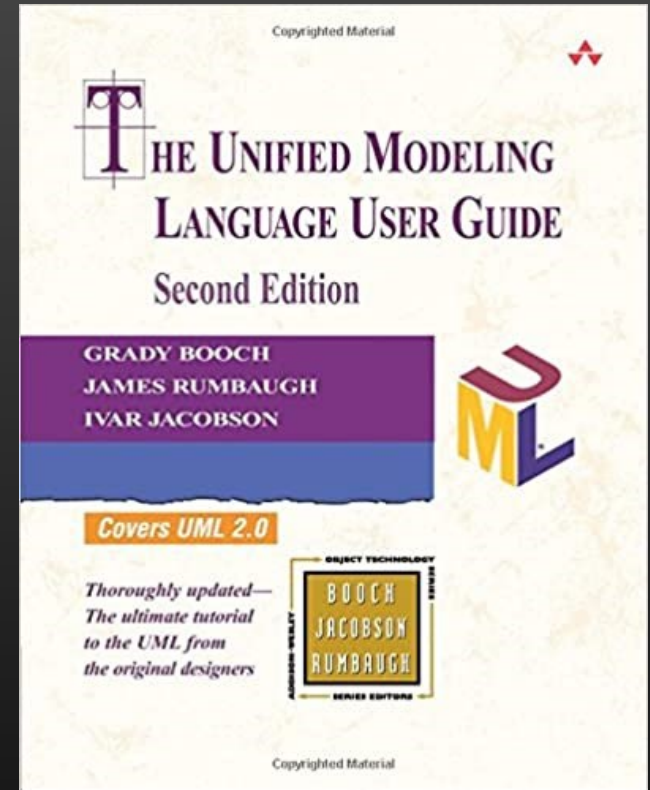
### Benefícios

Promover a comunicação mais clara e sucinta

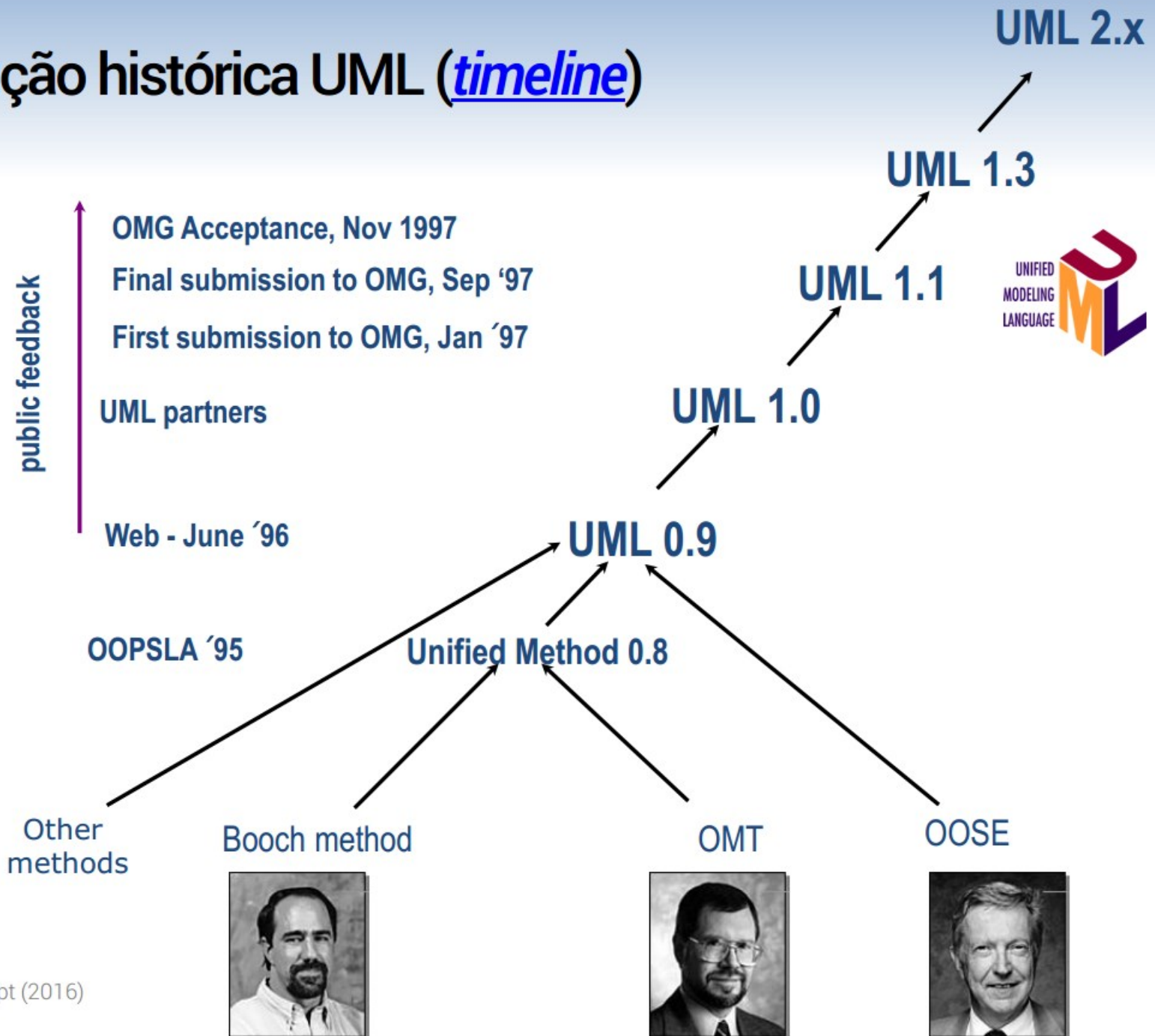
Manter o desenho (planeamento) e a implementação (construção) coerentes

Mostrar ou esconder diferentes níveis de detalhe, conforme apropriado

Pode suportar, em parte, processos de construção automática (gerar a solução a partir do modelo)



# Evolução histórica UML (timeline)



# *"Three Amigos"*

I. Jacobson: thinking in use-cases...



G. Booch: thinking in OO development process...

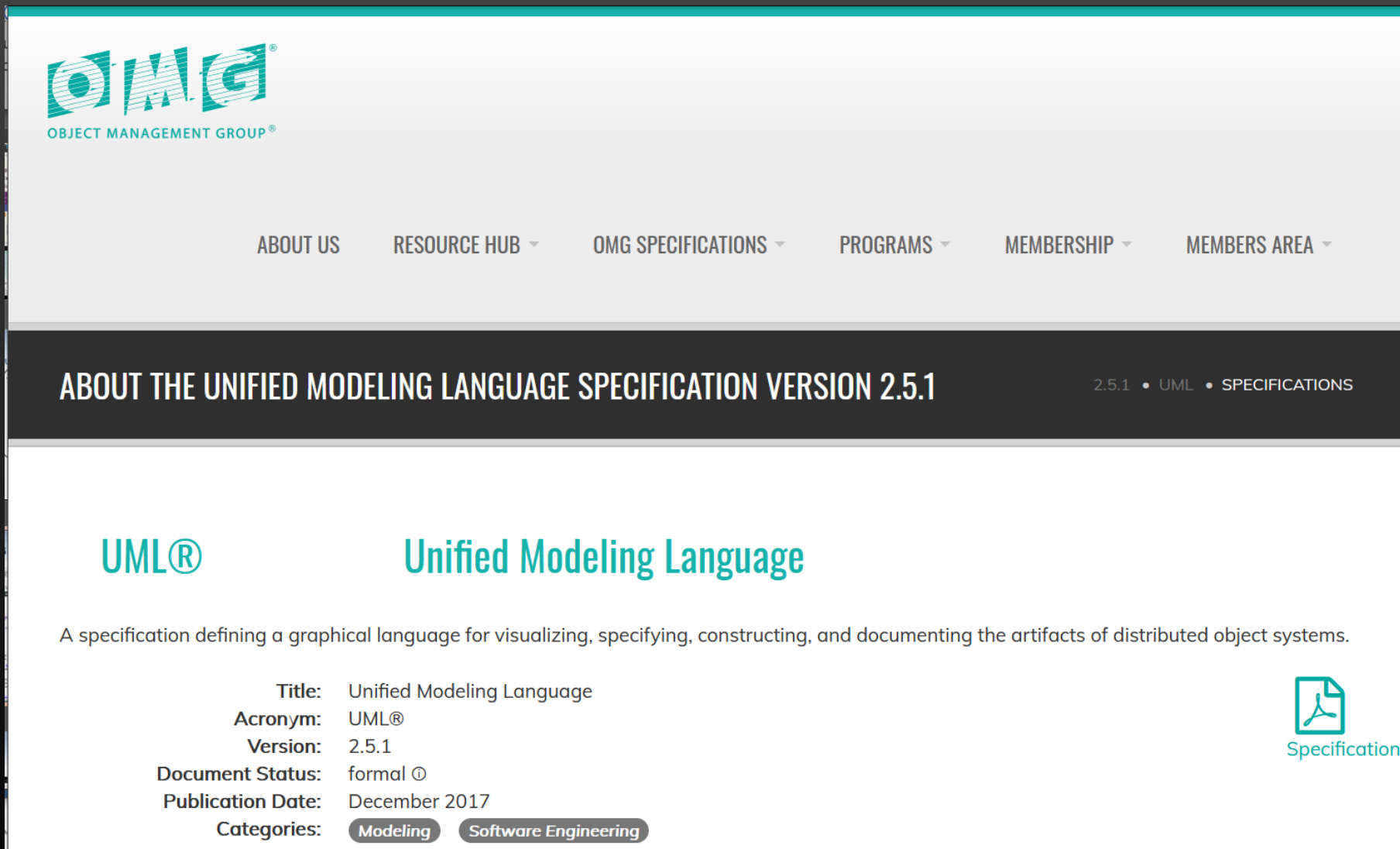


J. Rumbauch: thinking in classes modeling...



The UML effort started officially in October 1994, with the version 0.8 draft being released in October 1995. The Three Amigos, as they are sometimes referred to (Booch, Rumbaugh, and Jacobson) had successfully unified semantics and notation, ultimately meaning that users could focus on their own work and worry less about the specifics of a given method.

# UML é uma especificação do OMG



The screenshot shows the top portion of the OMG website. At the top left is the OMG logo with the text 'OBJECT MANAGEMENT GROUP®'. A navigation bar contains links: 'ABOUT US', 'RESOURCE HUB ▾', 'OMG SPECIFICATIONS ▾', 'PROGRAMS ▾', 'MEMBERSHIP ▾', and 'MEMBERS AREA ▾'. Below this is a dark banner with the text 'ABOUT THE UNIFIED MODELING LANGUAGE SPECIFICATION VERSION 2.5.1' on the left and '2.5.1 • UML • SPECIFICATIONS' on the right. The main content area features the 'UML®' logo and the title 'Unified Modeling Language'. A paragraph describes it as 'A specification defining a graphical language for visualizing, specifying, constructing, and documenting the artifacts of distributed object systems.' Below this is a list of metadata: Title: Unified Modeling Language; Acronym: UML®; Version: 2.5.1; Document Status: formal ⓘ; Publication Date: December 2017; and Categories: 'Modeling' and 'Software Engineering' (each in a blue pill-shaped button). On the right side of the metadata is a PDF icon and the word 'Specification'.

**OMG**  
OBJECT MANAGEMENT GROUP®


ABOUT US   RESOURCE HUB ▾   OMG SPECIFICATIONS ▾   PROGRAMS ▾   MEMBERSHIP ▾   MEMBERS AREA ▾

**ABOUT THE UNIFIED MODELING LANGUAGE SPECIFICATION VERSION 2.5.1**   2.5.1 • UML • SPECIFICATIONS

**UML®**   **Unified Modeling Language**

A specification defining a graphical language for visualizing, specifying, constructing, and documenting the artifacts of distributed object systems.

**Title:** Unified Modeling Language  
**Acronym:** UML®  
**Version:** 2.5.1  
**Document Status:** formal ⓘ  
**Publication Date:** December 2017  
**Categories:** **Modeling** **Software Engineering**

 Specification

# Também reconhecida como um standard internacional ISO



The screenshot shows the ISO Store website interface. At the top, there is a navigation bar with the ISO logo on the left and links for 'Standards', 'About us', 'Standards Development', 'News', and 'Store'. To the right of these links are language options 'Français | Русский' and a 'Members area' link. Below the navigation bar is a search bar labeled 'Search ISO'. The main content area displays the breadcrumb path: 'ISO Store > Store > Standards catalogue > By TC > JTC 1 Information technology > SC 7'. The product title is 'ISO/IEC 19505-1:2012' with a small orange icon. Below the title is the subtitle 'Information technology -- Object Management Group Unified Modeling Language (OMG UML) -- Part 1: Infrastructure'. There is an 'Abstract' section on the left and a 'Preview ISO/IEC 19505-1:2012' button. On the right, there is a table with two columns: 'FORMAT' and 'LANGUAGE'. The 'FORMAT' column has two rows: 'PDF' (selected with a checkmark) and 'PAPER'. The 'LANGUAGE' column has two rows, both with 'English' selected in a dropdown menu.

ISO Store > Store > Standards catalogue > By TC > JTC 1 Information technology > SC 7

## ISO/IEC 19505-1:2012<sup>®</sup>

Information technology -- Object Management Group Unified Modeling Language (OMG UML) -- Part 1: Infrastructure

### Abstract

[Preview ISO/IEC 19505-1:2012](#)

ISO/IEC 19505-1:2012 defines the Unified Modeling Language (UML), revision 2. The objective of UML is to provide system architects, software engineers, and software developers with tools for analysis, design, and implementation of software-based systems as well as for modeling business and similar processes.

FORMAT	LANGUAGE
<input checked="" type="checkbox"/> PDF	English
<input type="checkbox"/> PAPER	English



# Aplicações principais da UML

## Análise e desenho de sistemas de software

Estrutura e comportamento de sistemas baseados em software

- Elementos do modelo representam entidades do mundo do software

Especialmente adequada para o desenvolvimento por objetos (*object-oriented*)

## Domínio do problema (processos de trabalho,...)

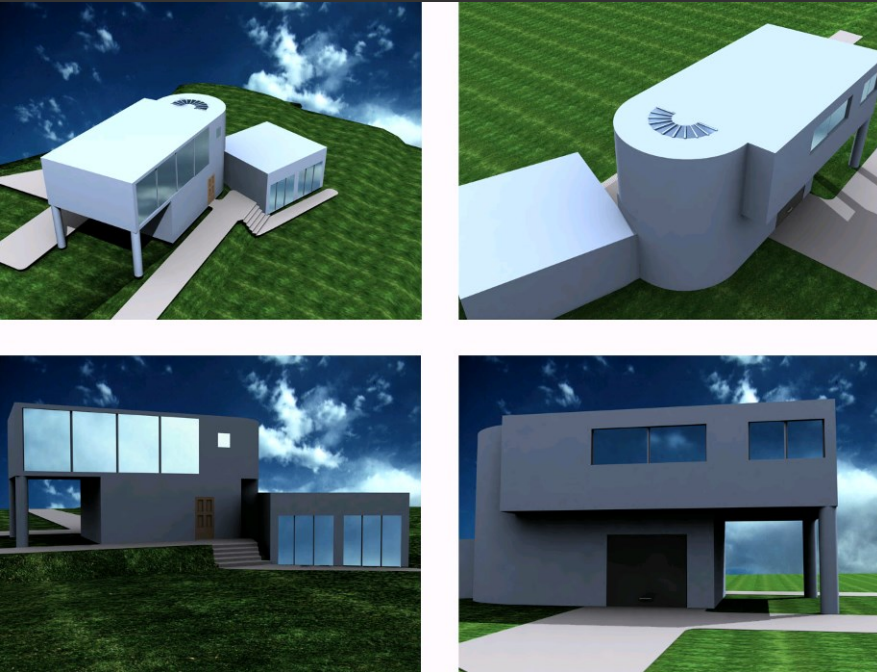
Especificar ou documentar o domínio de aplicação/negócio

- Elementos do modelo representam entidades do negócio

Não implica ou assume uma implementação em software



Não há uma vista única, mas várias e complementares



Para que serve o sistema?

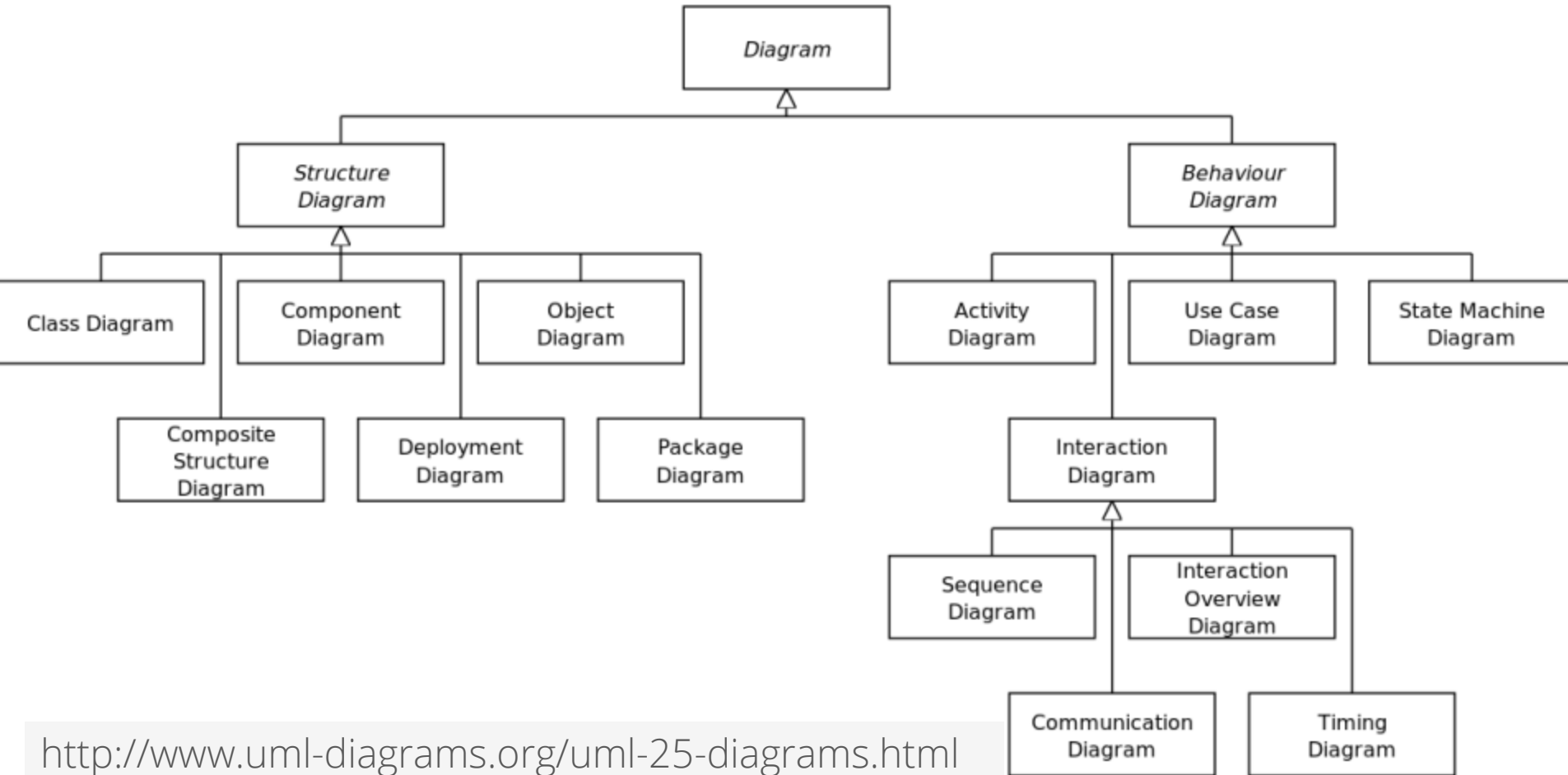
Quais são as estruturas de informação?

Decomposição funcional de atividades complexas

Visualizar a organização do software em partes e as suas interações

Etc.

# Diagramas da UML 2.x

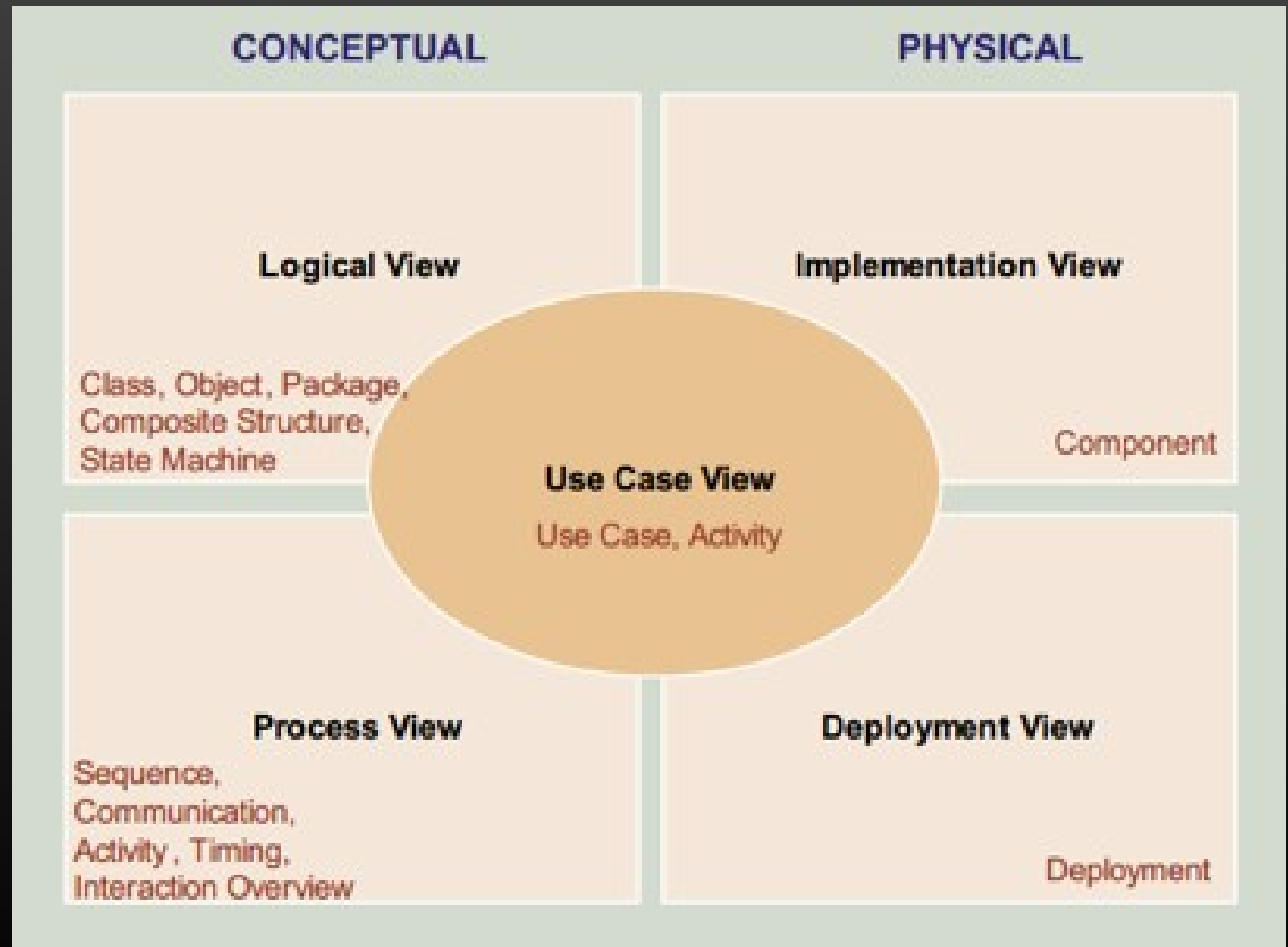




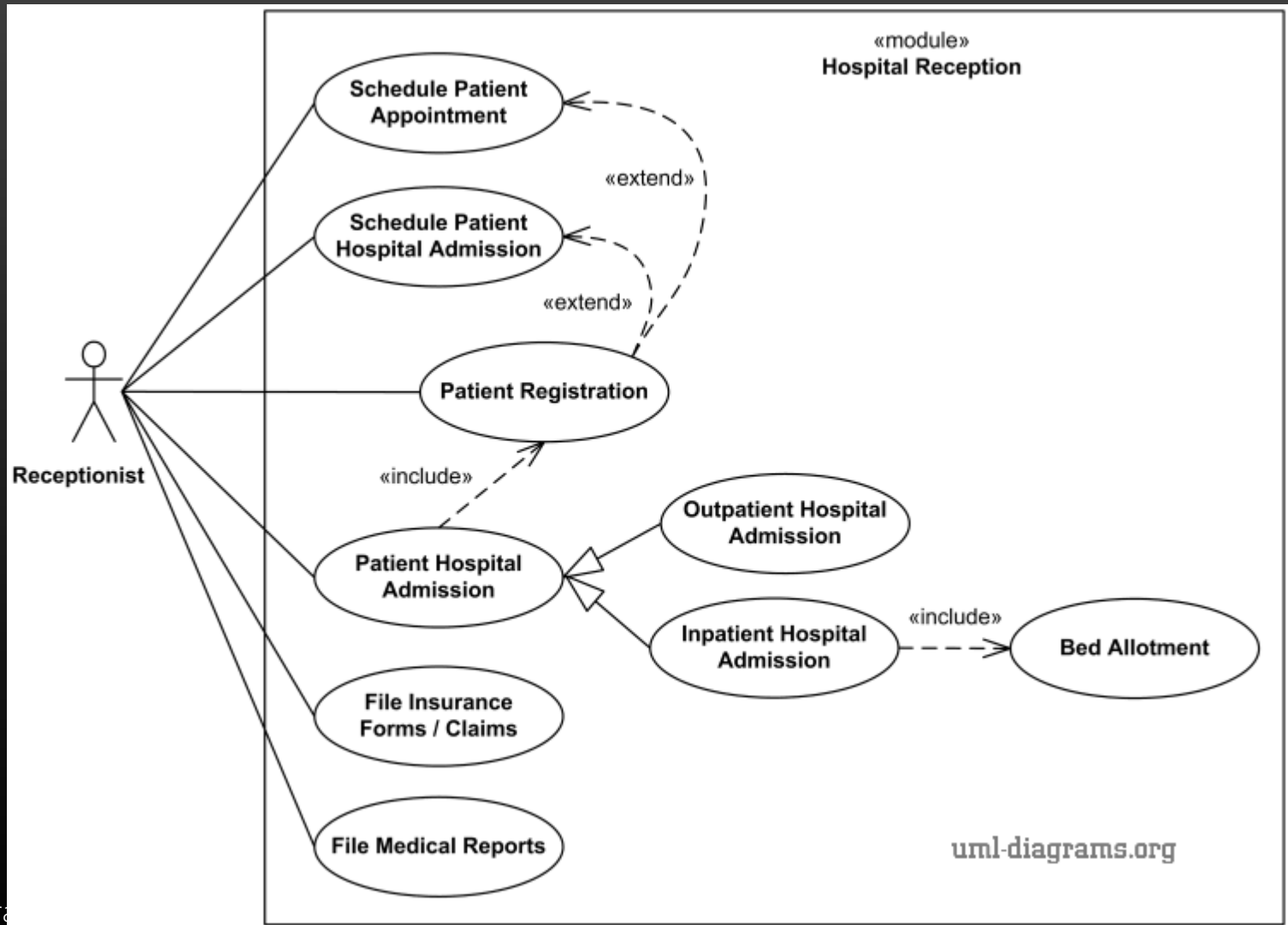
# APLICAÇÃO DA UML AO LONGO DO PROCESSO DE DESENVOLVIMENTO

ILÍDIO OLIVEIRA [ico@ua.pt](mailto:ico@ua.pt)  
v2017-06-02

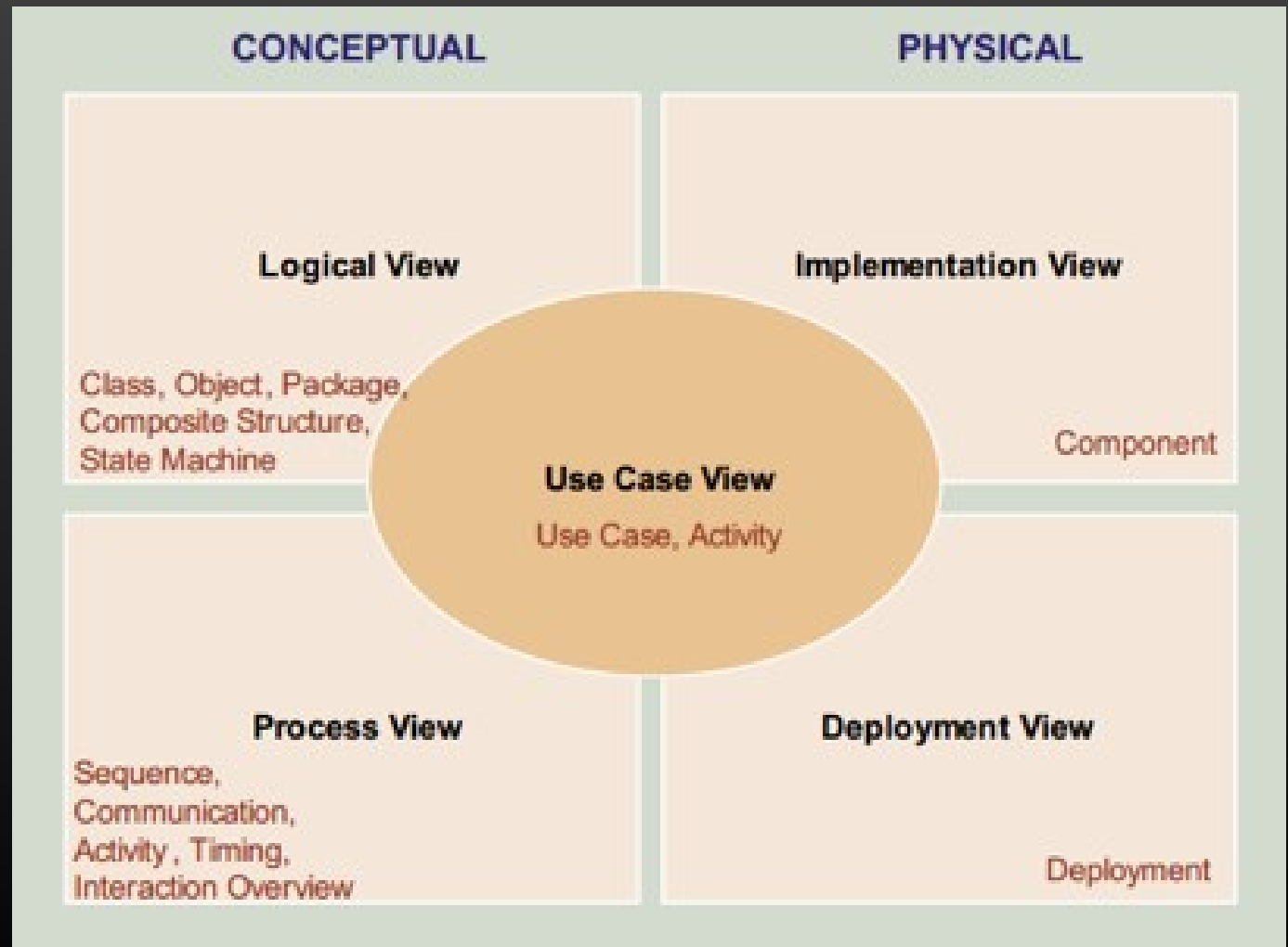
# Diversos diagramas para abranger diferentes perspectivas de análise



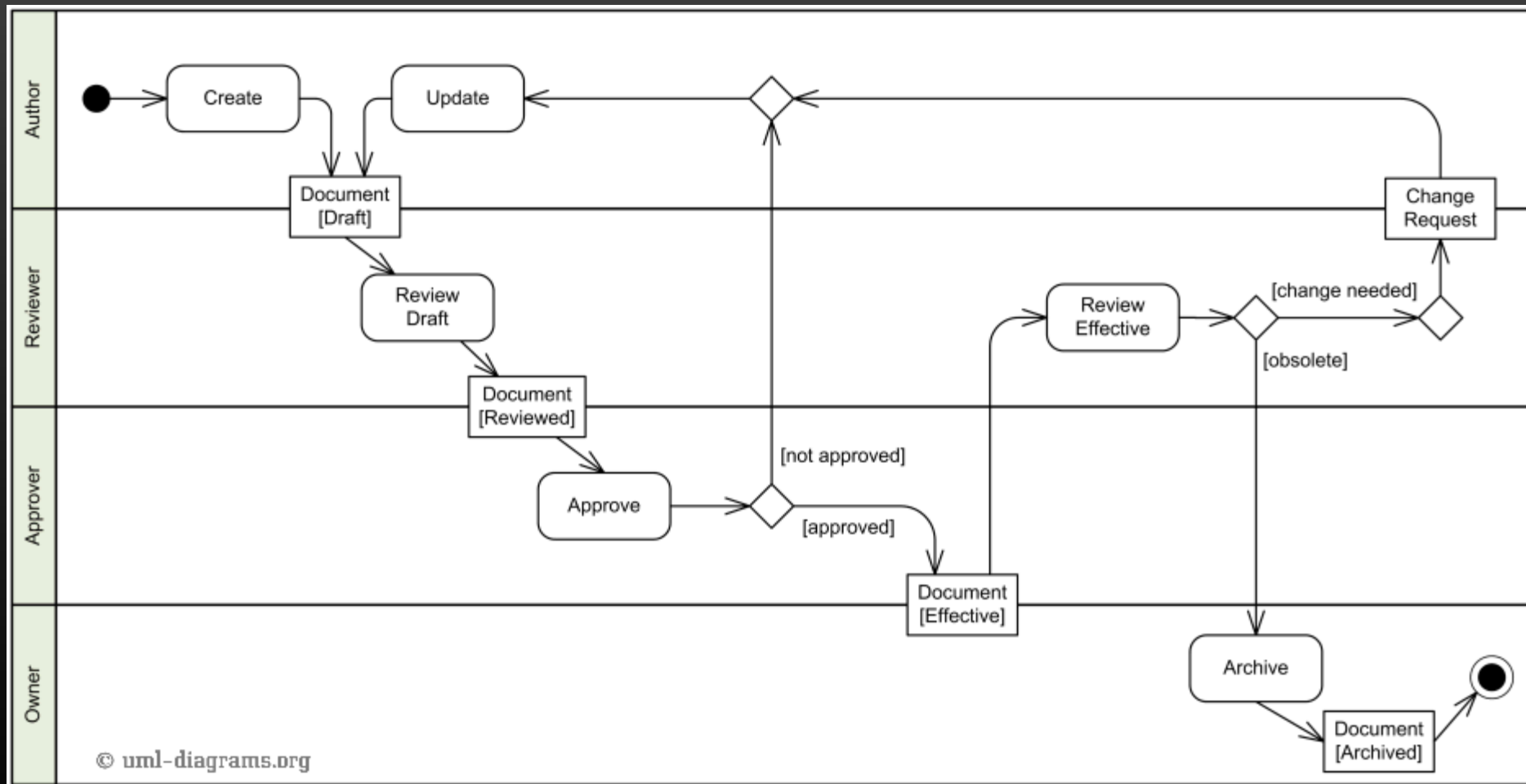
# CaU do Sistema: organizar a funcionalidade do sistema em episódios de utilização



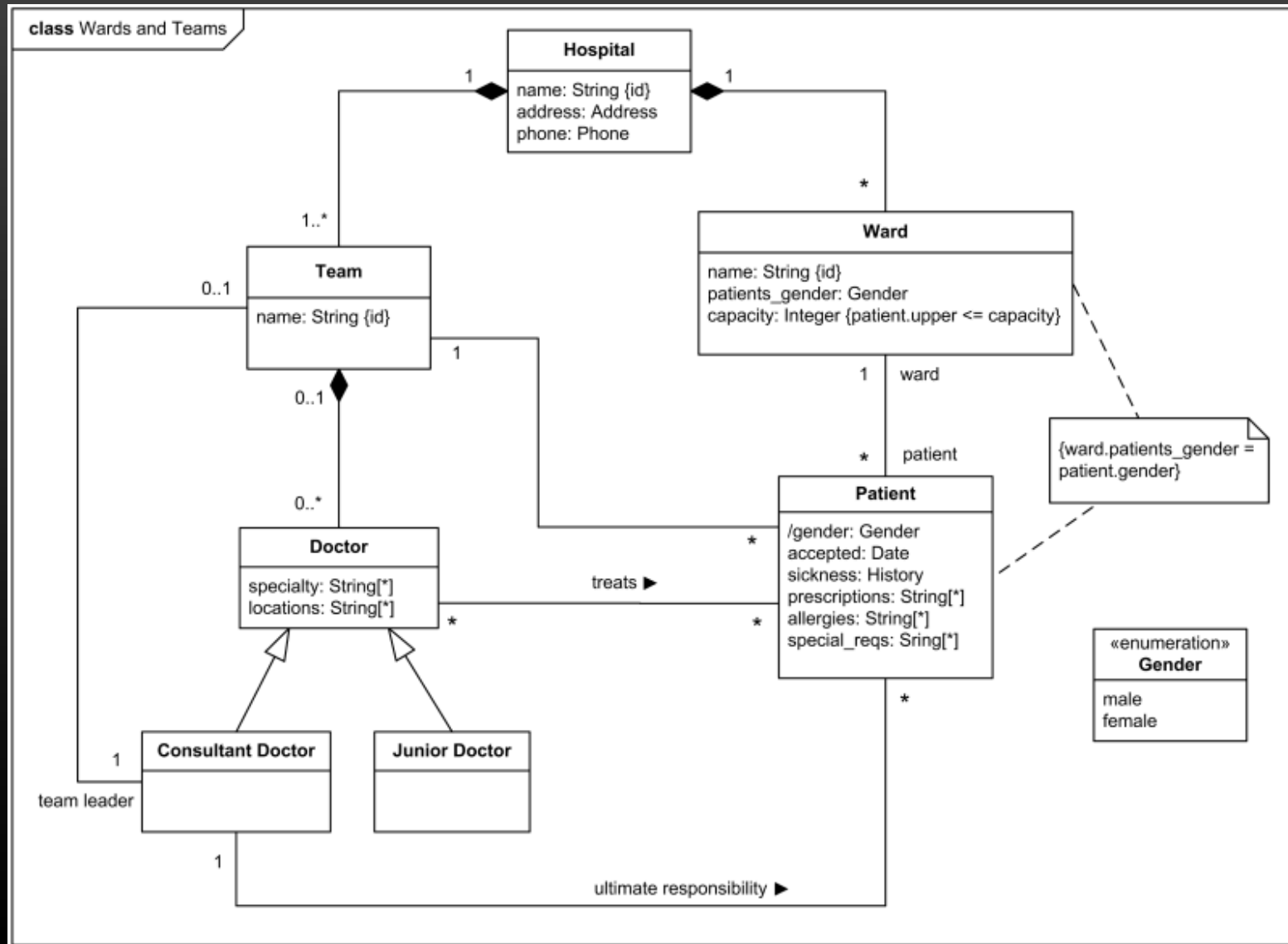
# Diversos diagramas para abranger diferentes perspectivas de análise



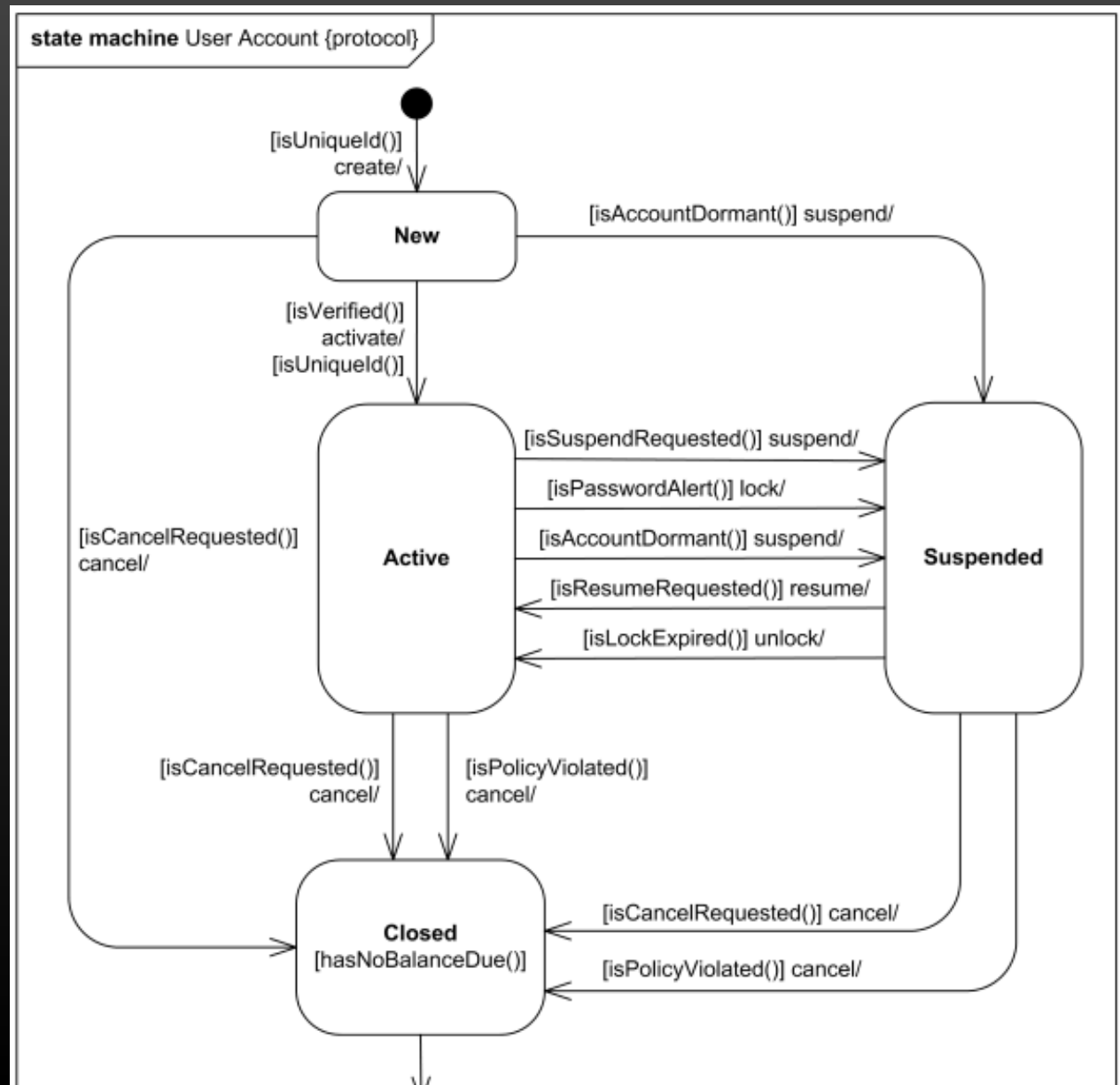
# Diagramas de atividades para explicar procedimentos do domínio



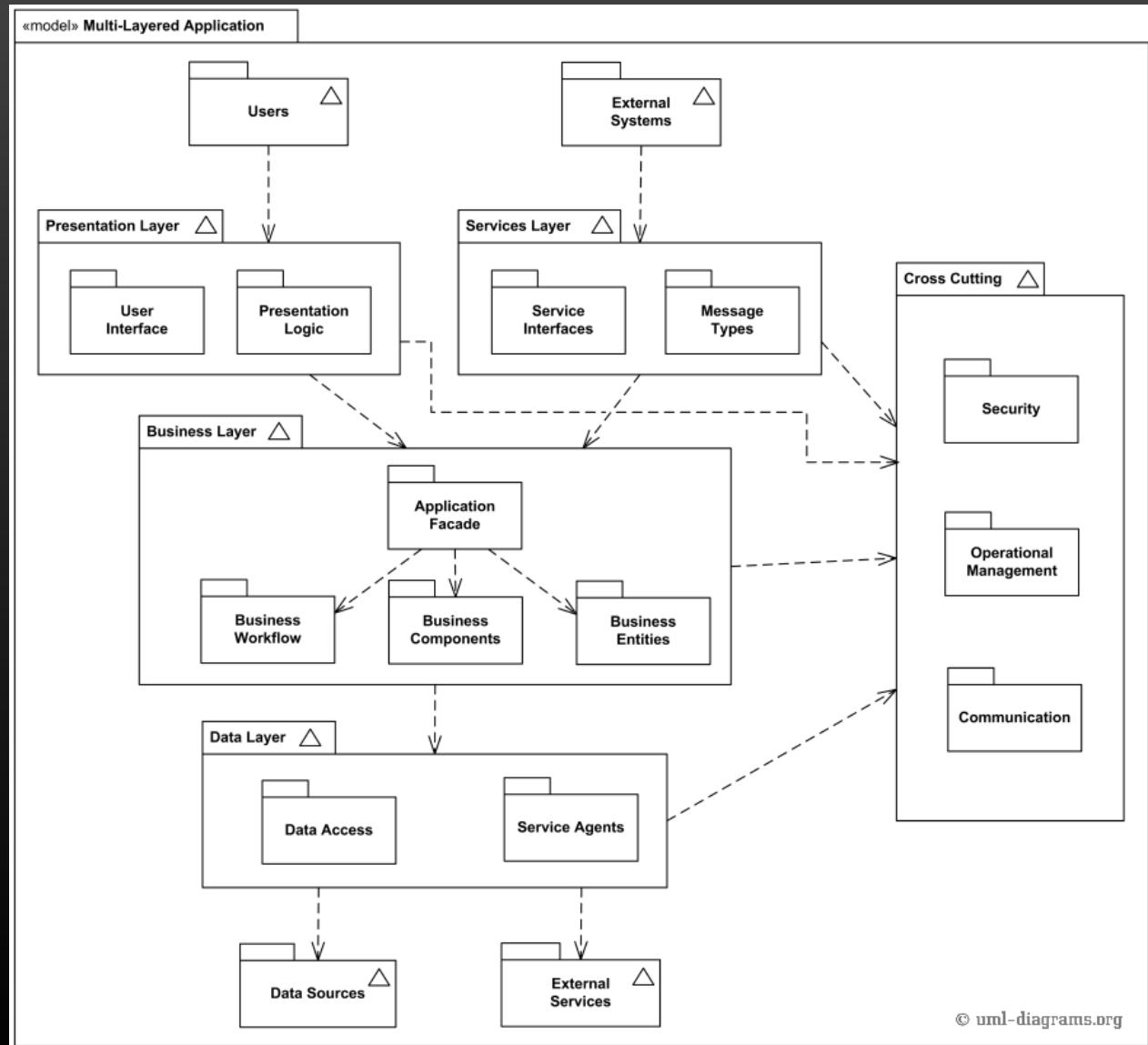
Classes para representar os conceitos da área do problema (modelo do domínio)



# Máquina de estados de entidades/objetos

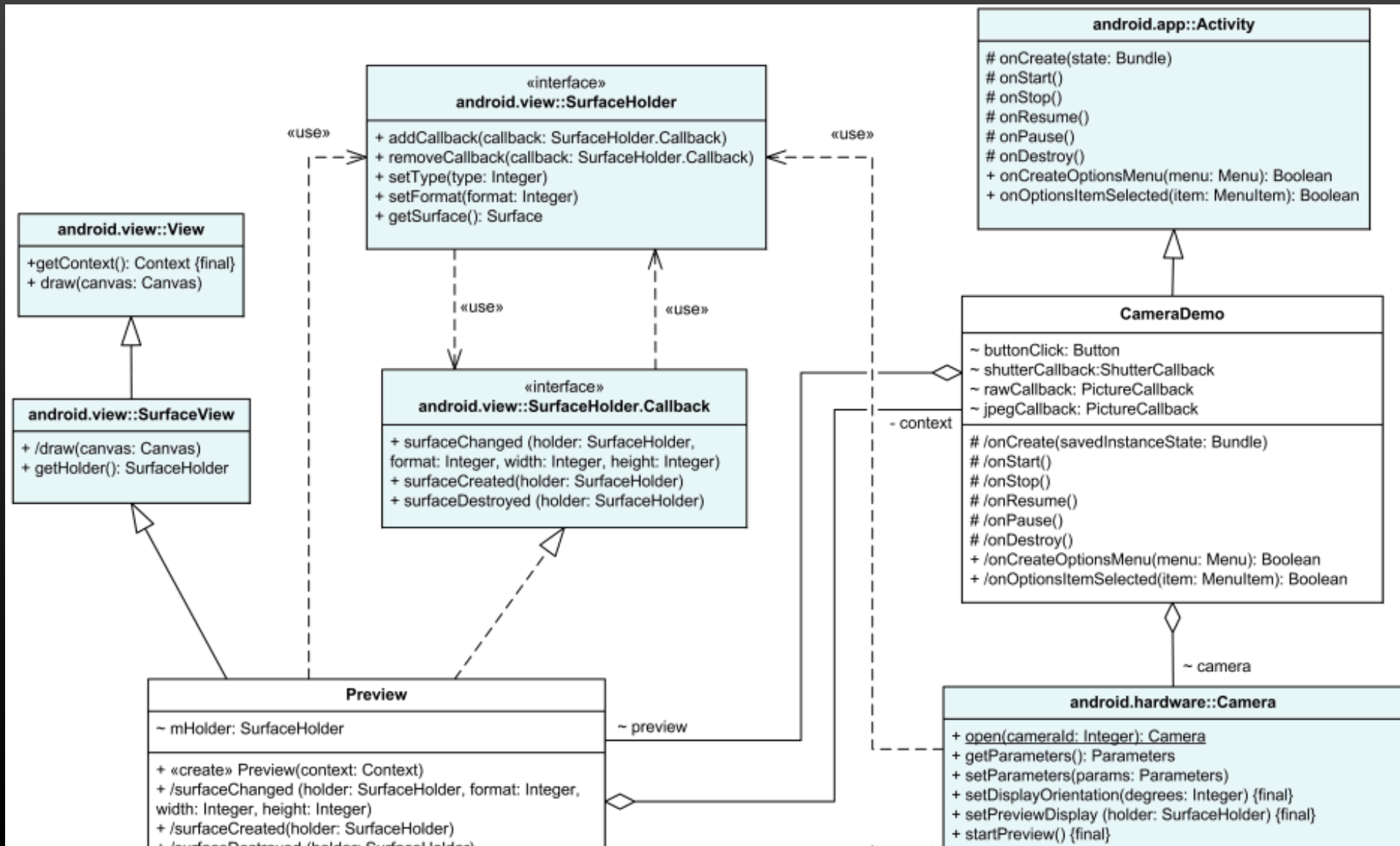


# Visualizar a arquitetura lógica com D. Pacotes

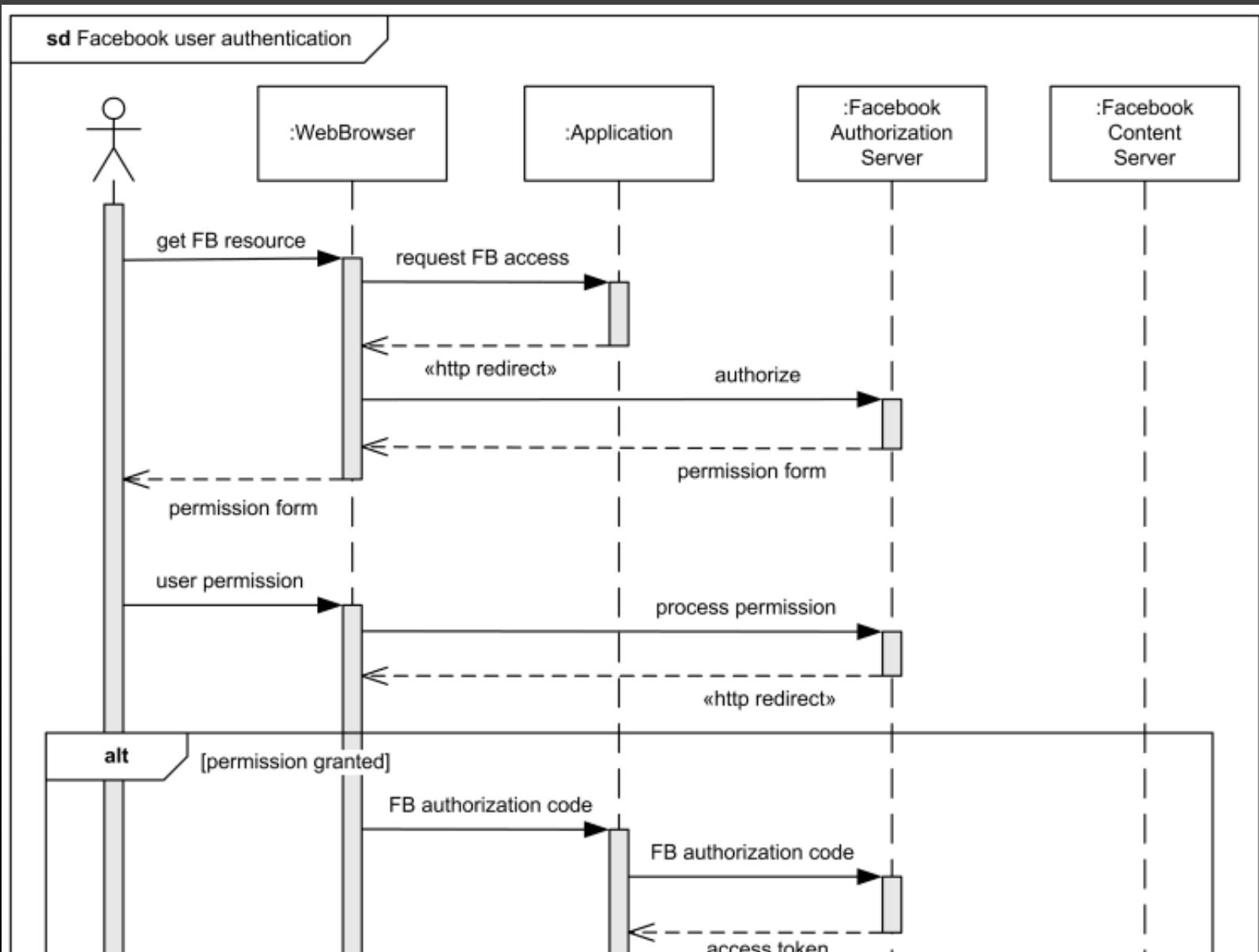




# Classes para visualizar objetos de um linguagem de programação

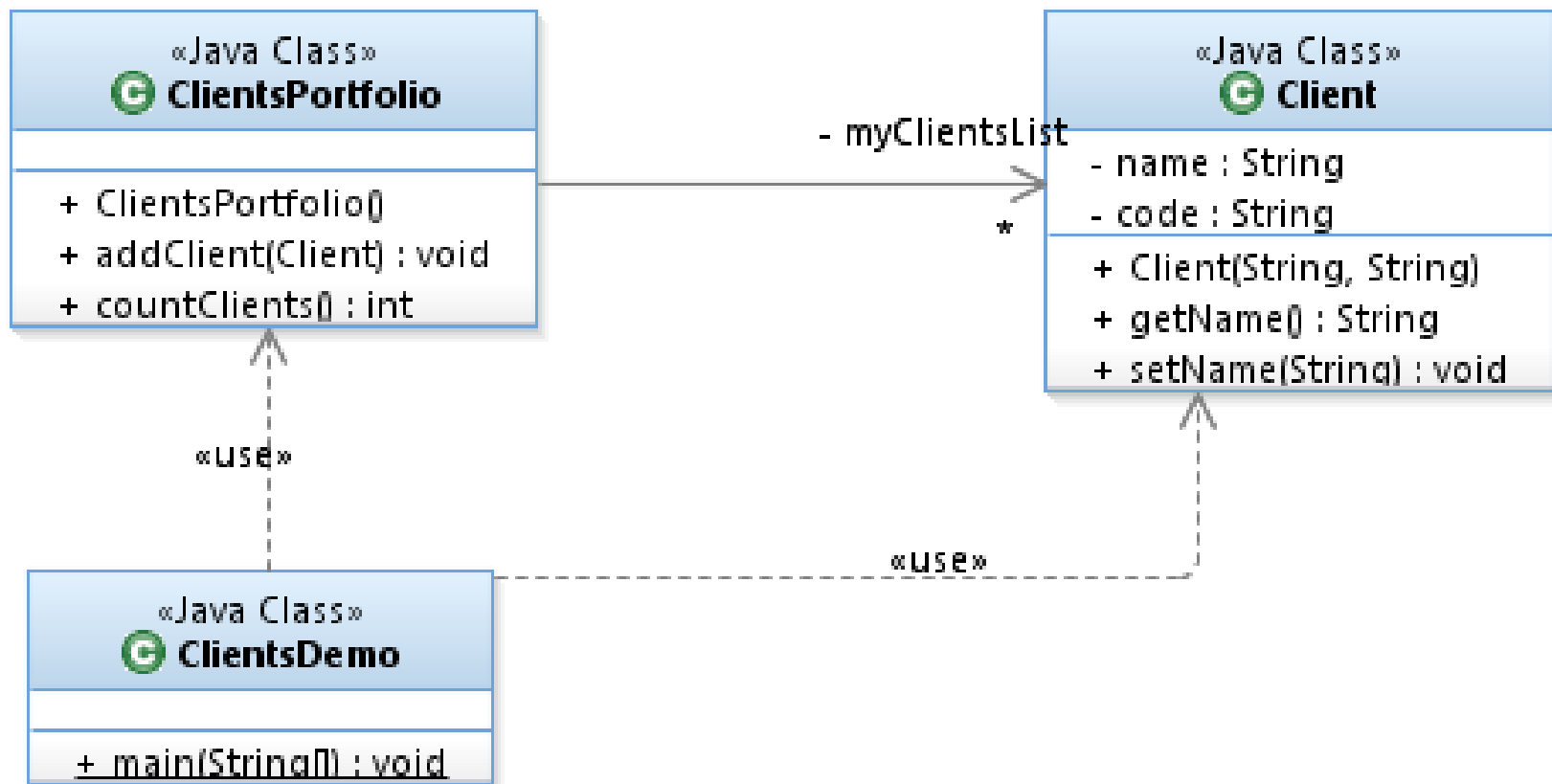


# Interações entre componentes do software

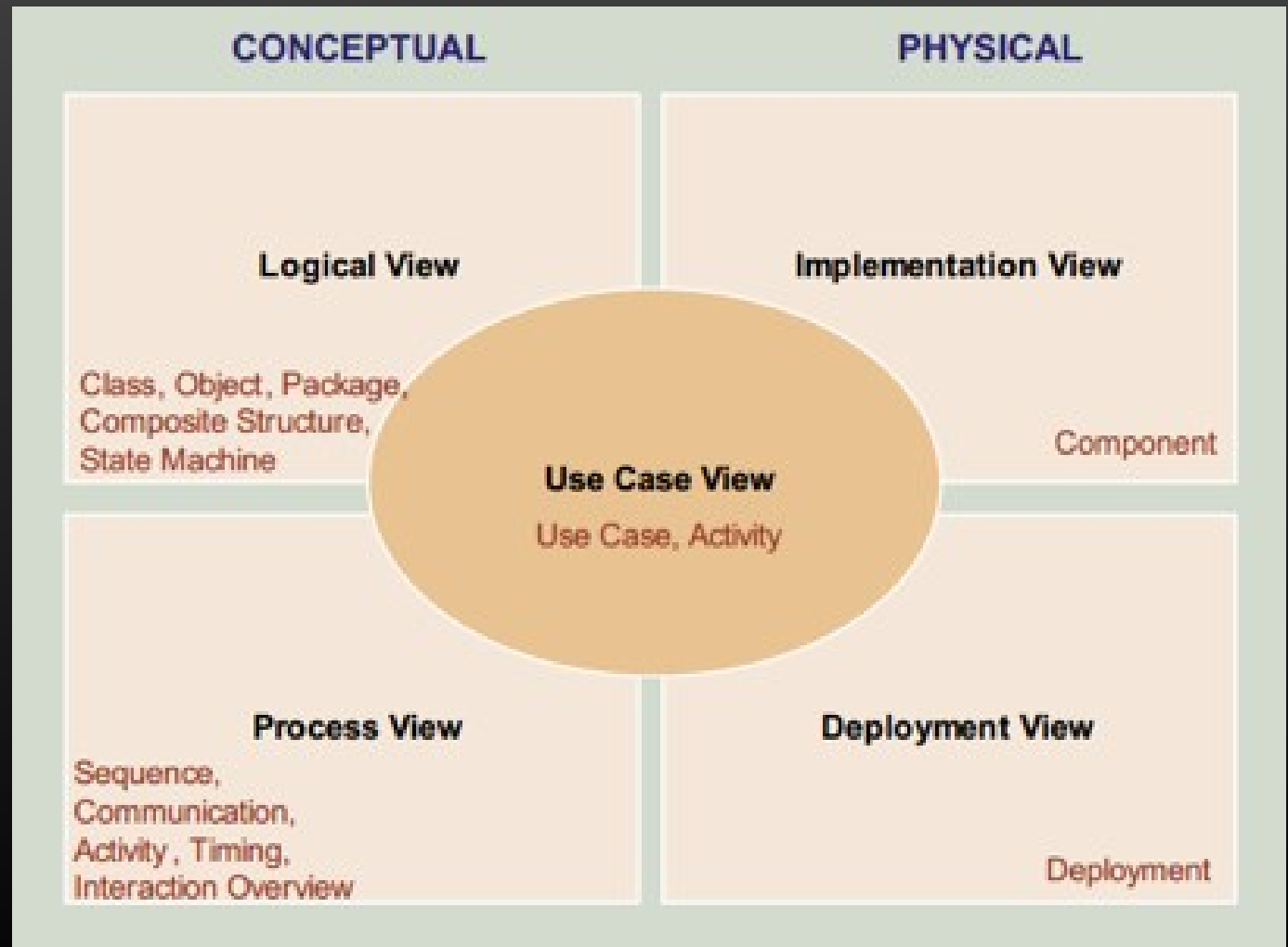


# Objetos em código

```
public class ClientsPortfolio {  
    private ArrayList<Client> myClientsList;  
  
    public ClientsPortfolio() {  
        myClientsList =new ArrayList<>();  
    }  
    public void addClient(Client newClient) {  
        this.myClientsList.add(newClient);  
    }  
    public int countClients() {  
        return this.myClientsList.size();  
    }  
}
```



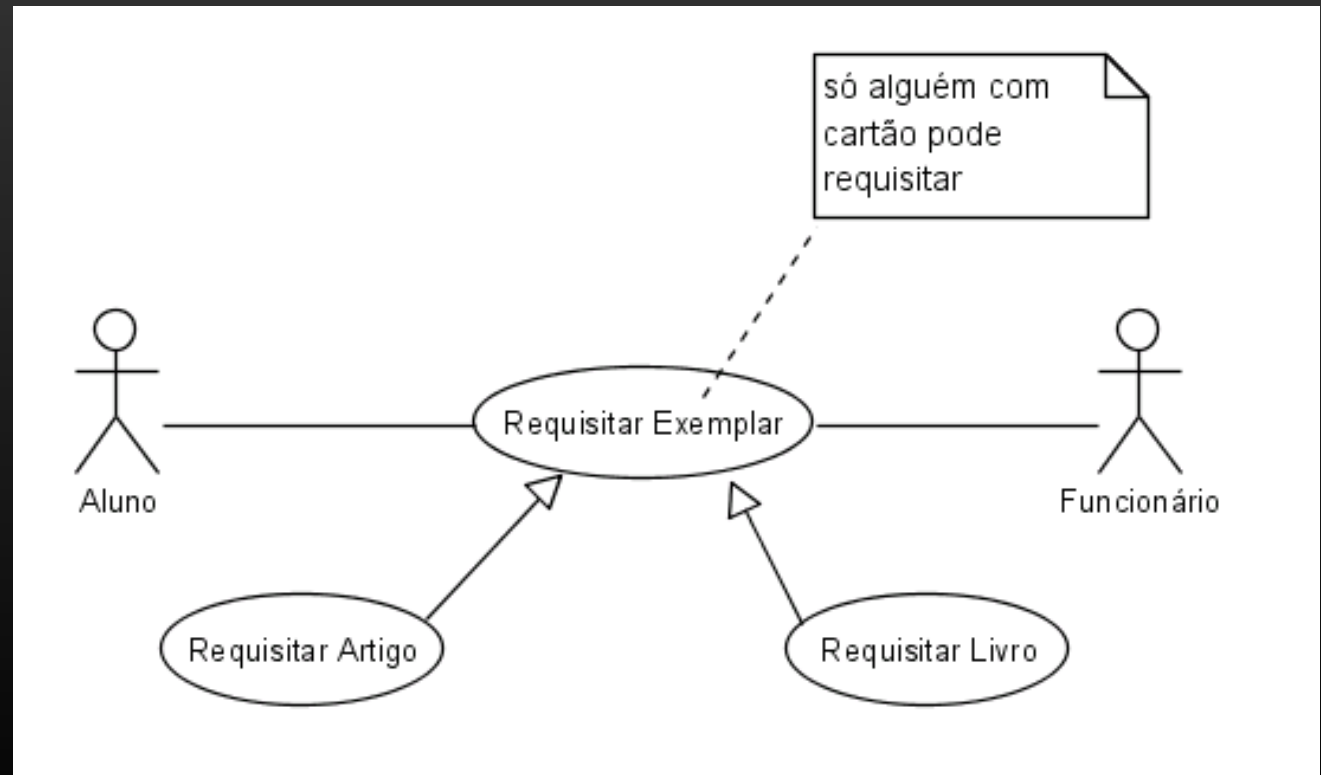
# Diversos diagramas para abranger diferentes perspectivas de análise



# Elementos comuns

# Anotações

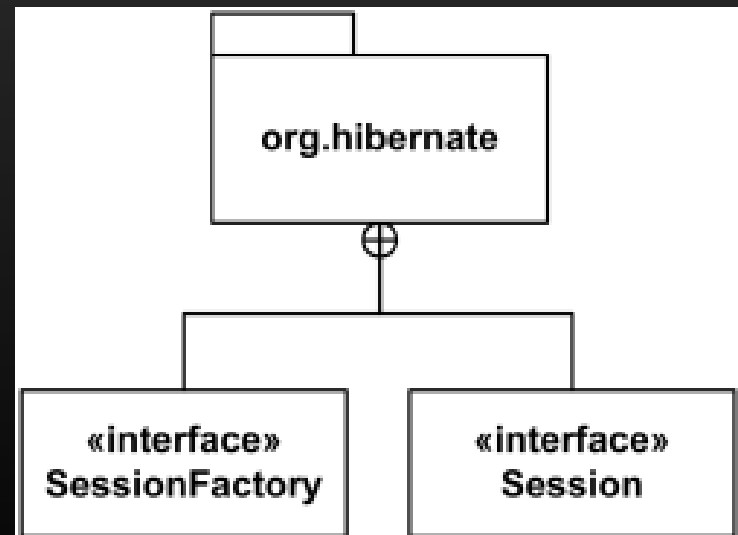
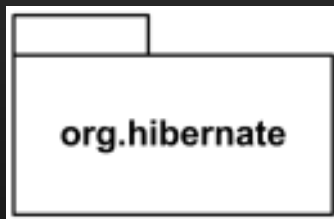
Um comentário que pode ser usado para anotar qualquer elemento



# Pacotes

um mecanismo para dividir um modelo em partes

serve como mecanismo genérico para fazer agrupamentos

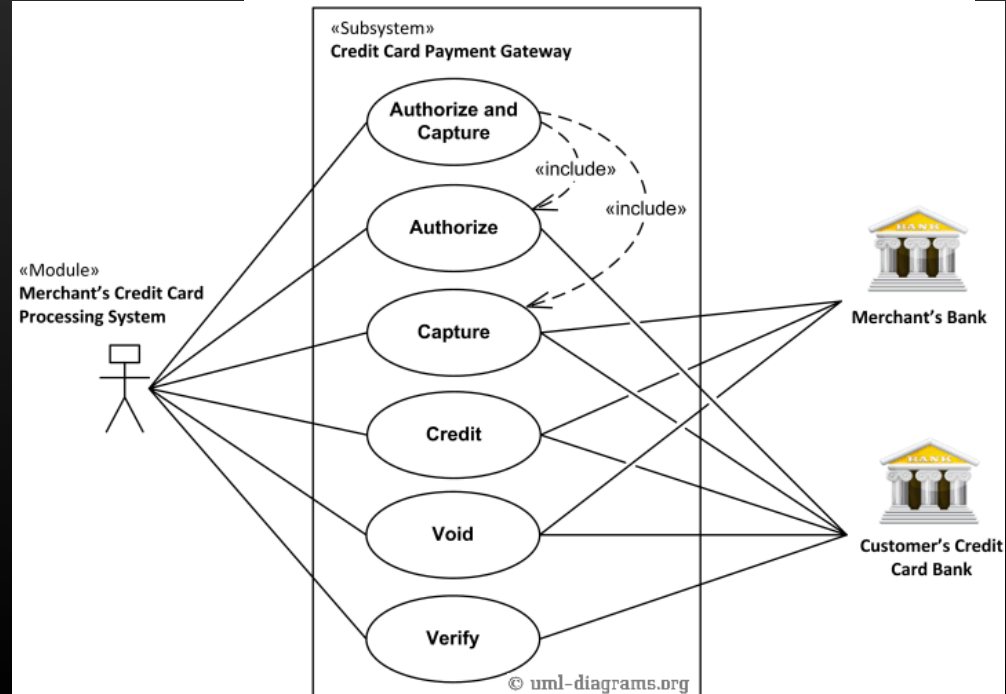
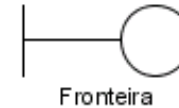
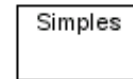
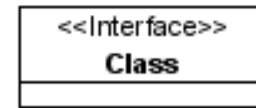




# Estereotipo (*stereotype*)

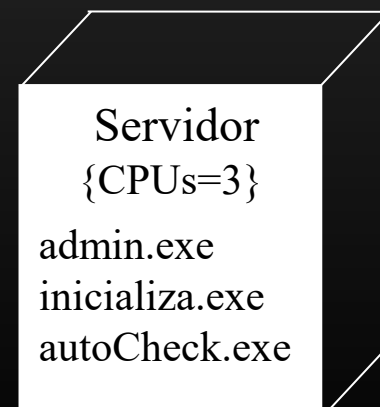
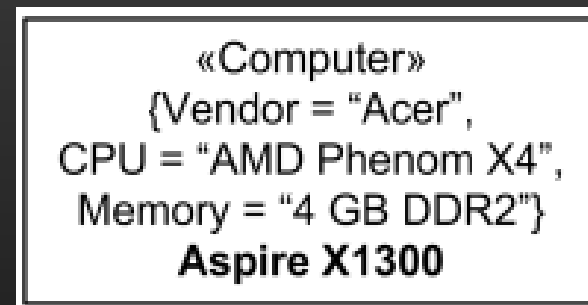
uma especialização da  
semântica de um elemento  
do modelação

marcada com «...» ou com  
a alteração da decoração



# Valores etiquetados (*tagged values*)

Estender elementos do modelo com uma linguagem “computável” (pares atributo/valor)

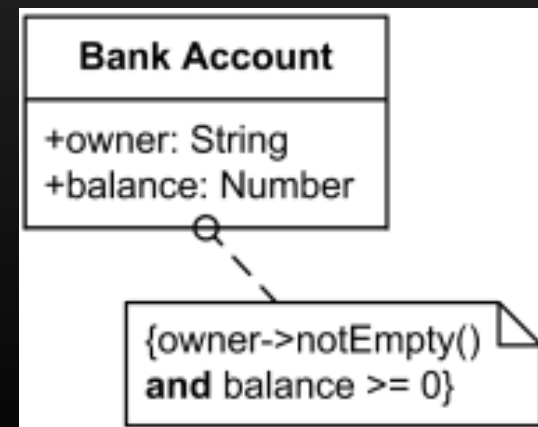
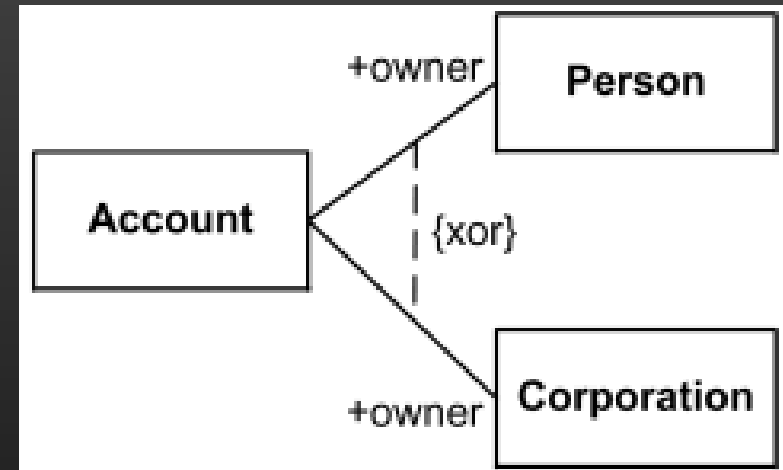
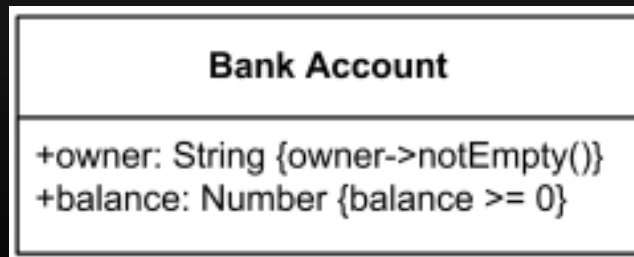


# Restrições

Adicionar regras ao modelo  
ou condicionar a sua  
interpretação

condição ou restrição  
relacionada com um ou mais  
elementos

Linguagem própria para  
declarar restrições (OCL)



# Readings & references

Core readings	Suggested readings
<ul style="list-style-type: none"><li data-bbox="150 411 705 464">• [Dennis15] – Chap. 1</li></ul>	<p data-bbox="962 411 1758 714">[LAR'12] Larman, C. (2012). <i>Applying UML and Patterns: An Introduction to Object Oriented Analysis and Design and Iterative Development</i>. Pearson Education.</p> <p data-bbox="962 785 1477 842">→ chap. 10, chap. 15.</p>