

47006- ANÁLISE E MODELAÇÃO DE SISTEMAS
1st Semester, 2018/19

Visual Modelling with UML

Ilídio Oliveira | 2020/10/09

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

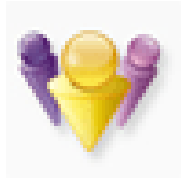
Read and create Activity Diagrams

The SDLC is applied through software development processes

What is in a process?



Core
Principles



Roles



Work Products



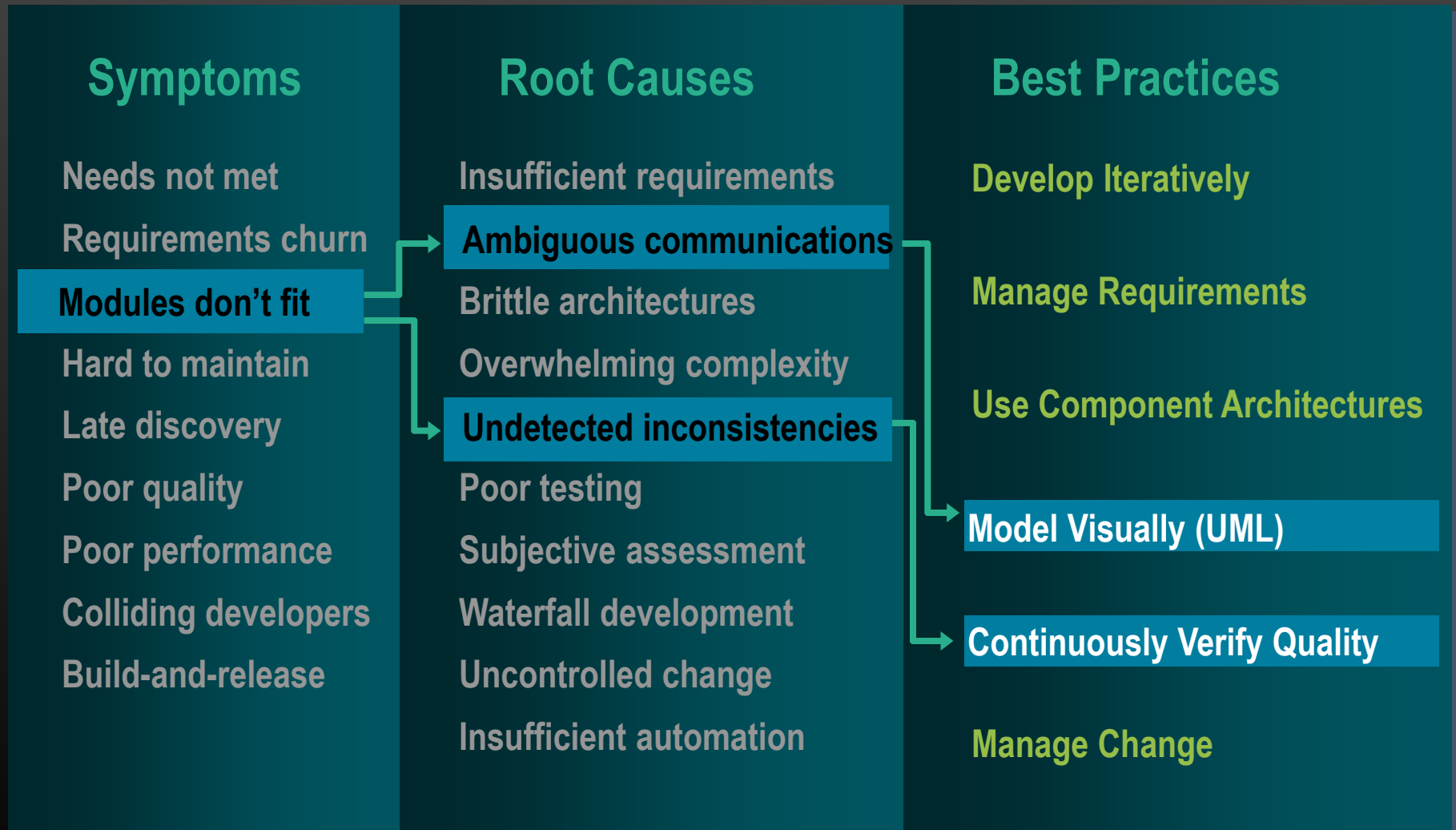
Disciplines



Lifecycle

http://sweet.ua.pt/ico/OpenUp/OpenUP_v1514/

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

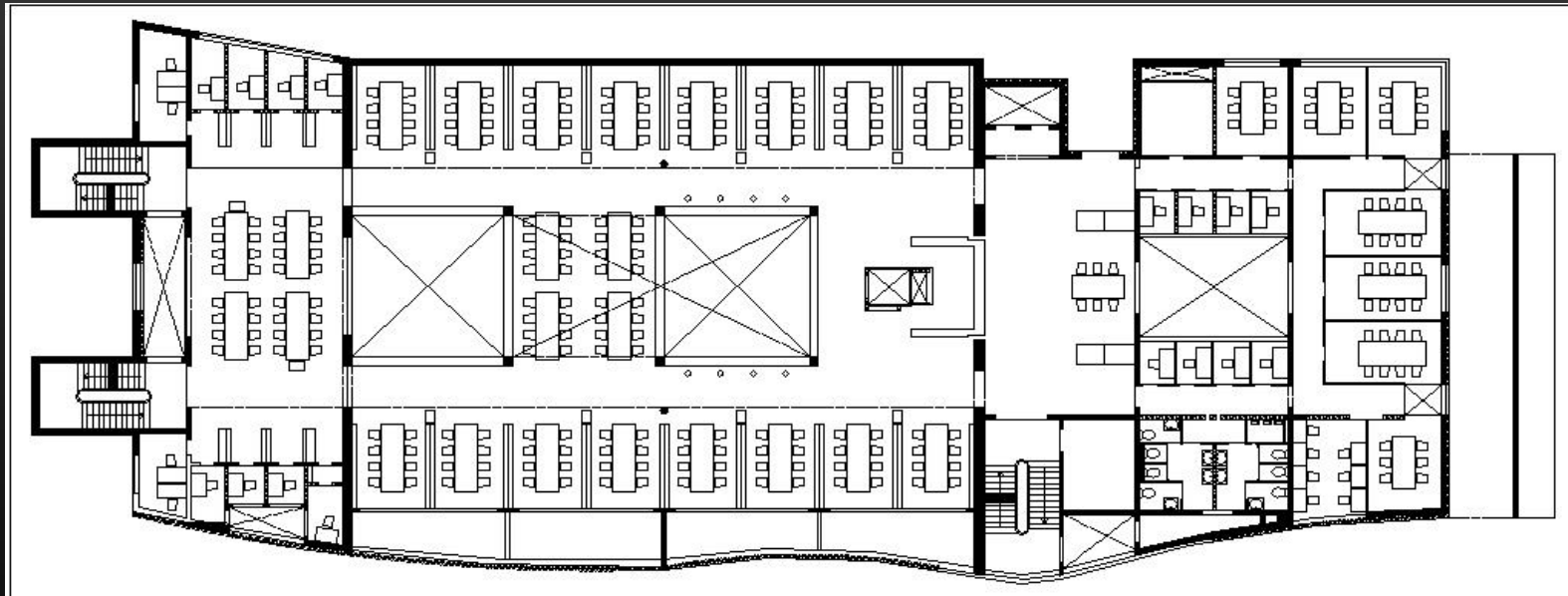
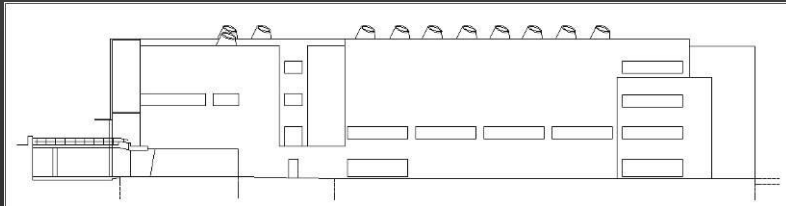
Organ



- 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

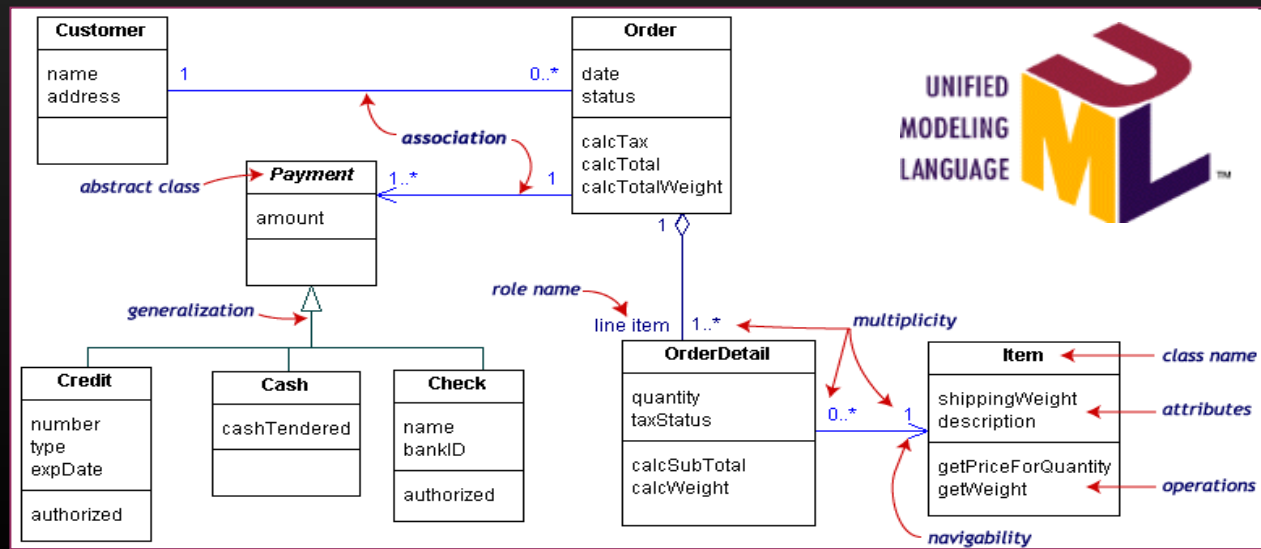
4 razões para usar modelos (G. Booch):

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

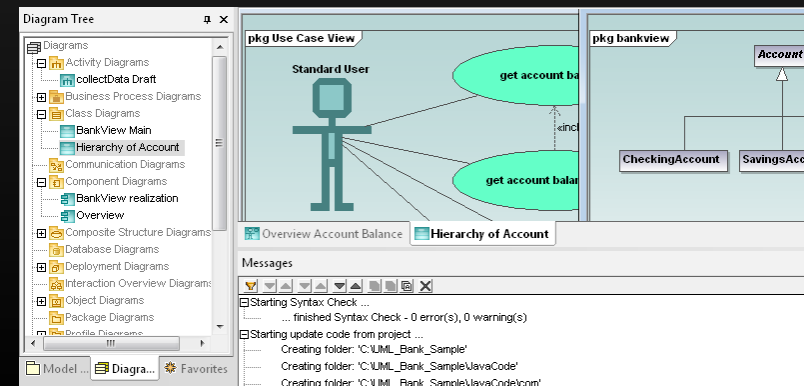
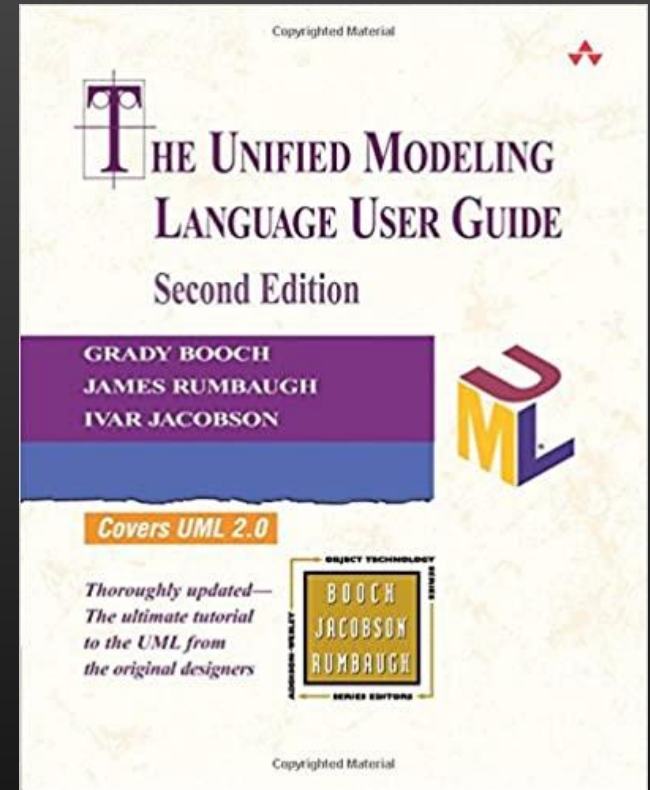
Benefícios

Promover a comunicação mais clara e sucinta

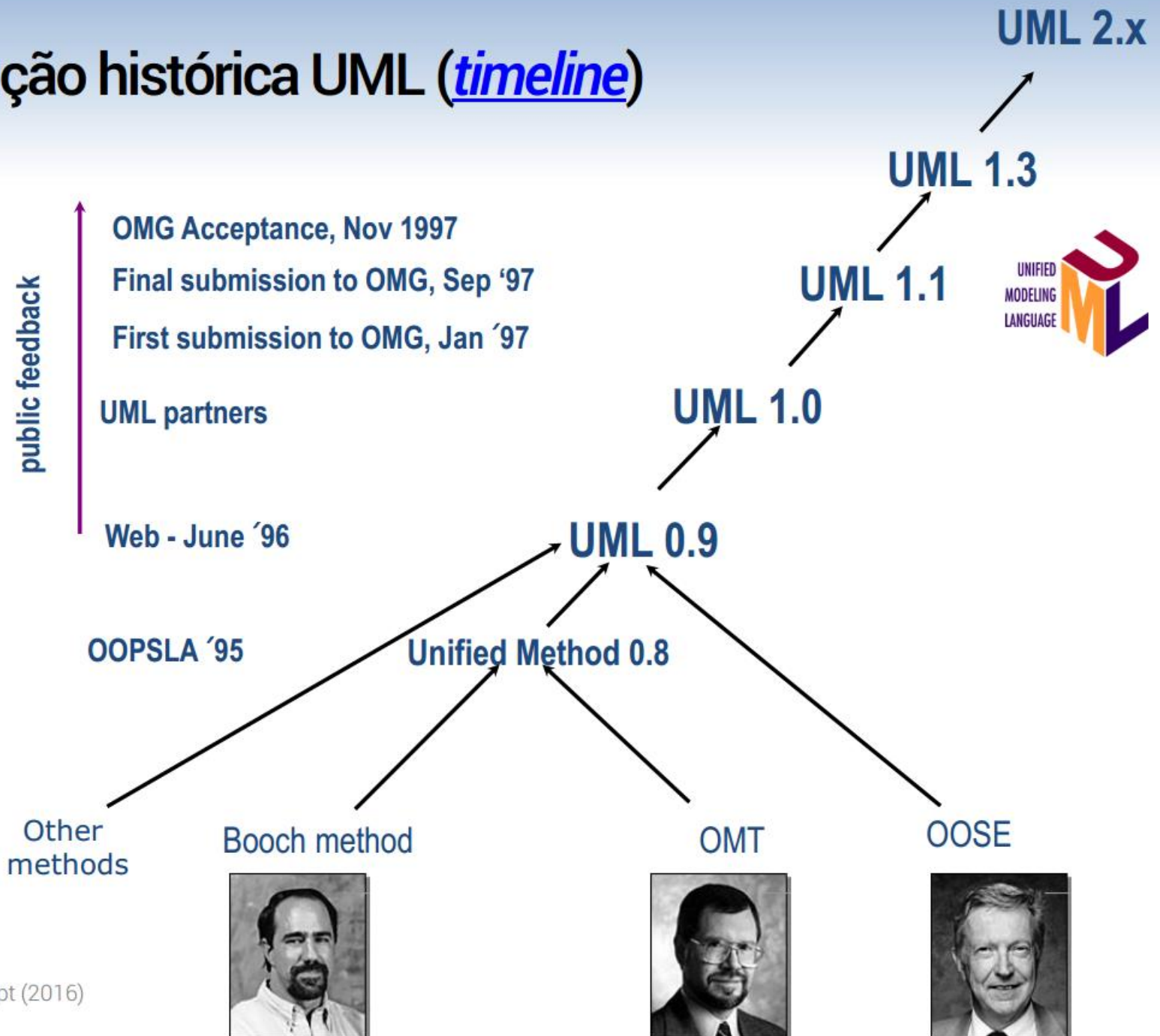
Manter o desenho (planeamento) e a implementação (construção) coerentes

Mostrar ou esconder diferentes níveis de detalhe, conforme apropriado

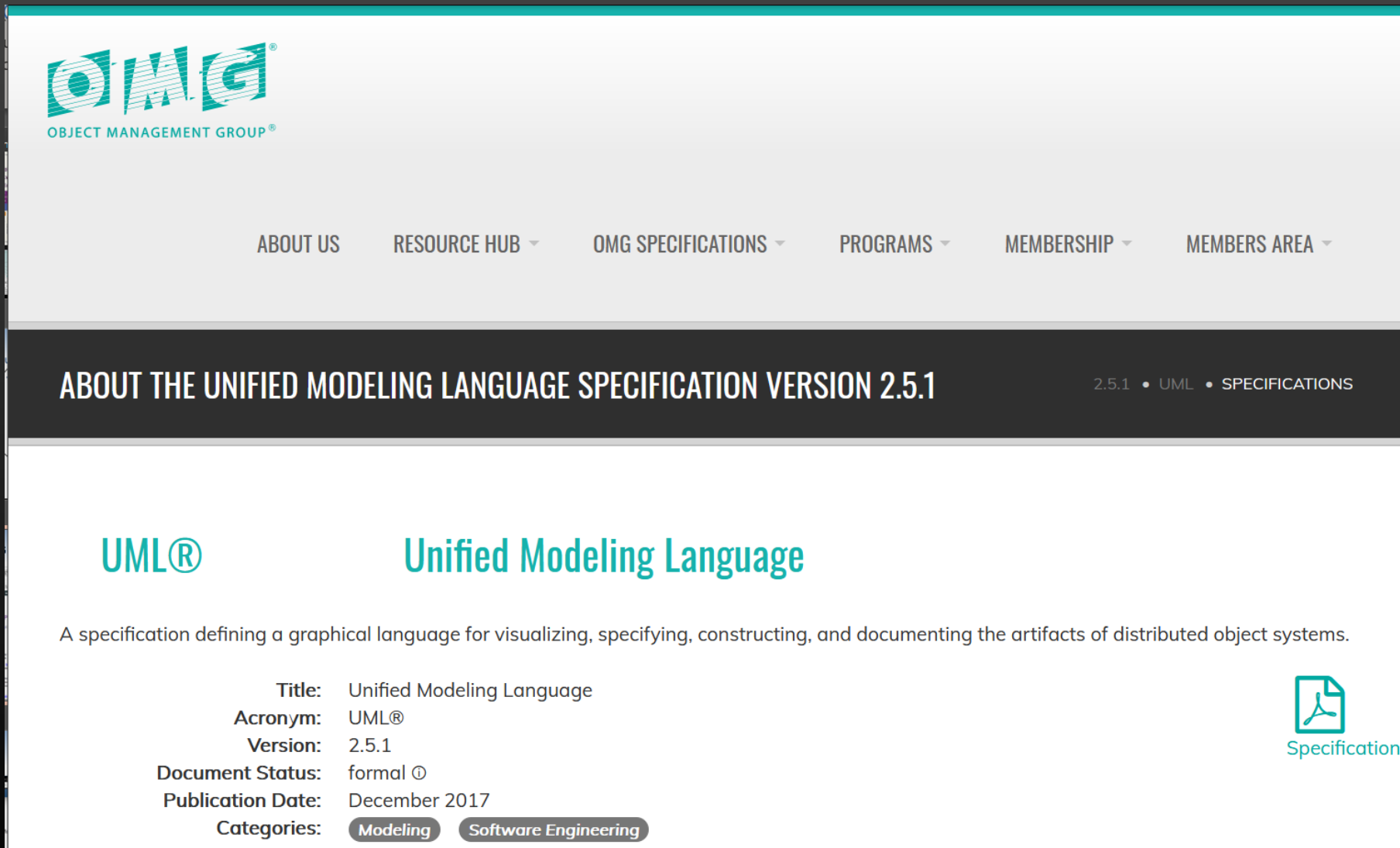
Pode suportar, em parte, processos de construção automática (gerar a solução a partir do modelo)



Evolução histórica UML (timeline)



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 UML as 'A specification defining a graphical language for visualizing, specifying, constructing, and documenting the artifacts of distributed object systems.' Below this is a table of metadata. To the right of the table is a PDF icon and the word 'Specification'.

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



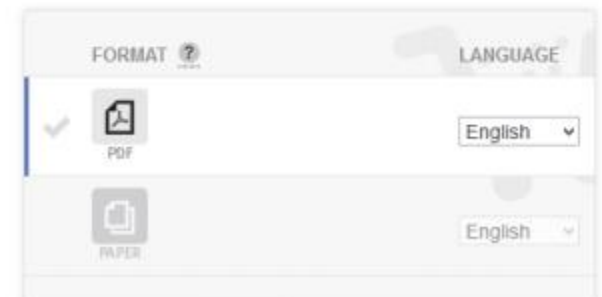
ISO/IEC 19505-1:2012[®]

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.



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

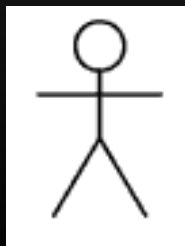
Especialmente adequada para o desenvolvimento por objetos (***object-oriented***)

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

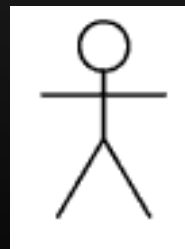
Especificar ou documentar o domínio de aplicação/negócio

- Elementos do modelo representam entidades do negócio

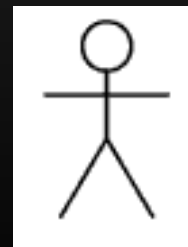
Não implica ou assume uma implementação em software



Analista



Programador



Arquiteto

Não há uma vista única, mas várias e complementares

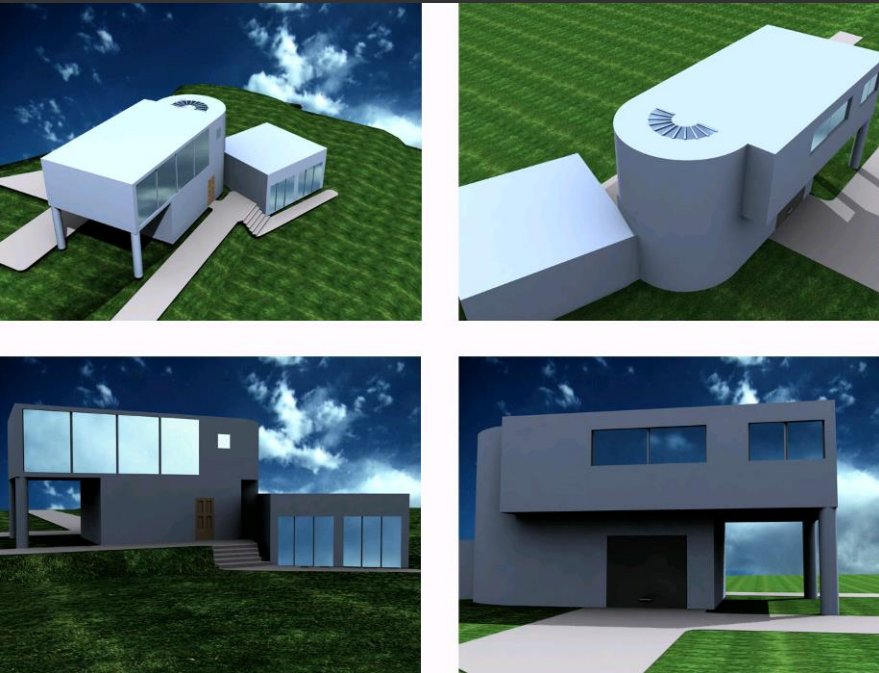
Para que serve o sistema?

Quais são as estruturas de informação?

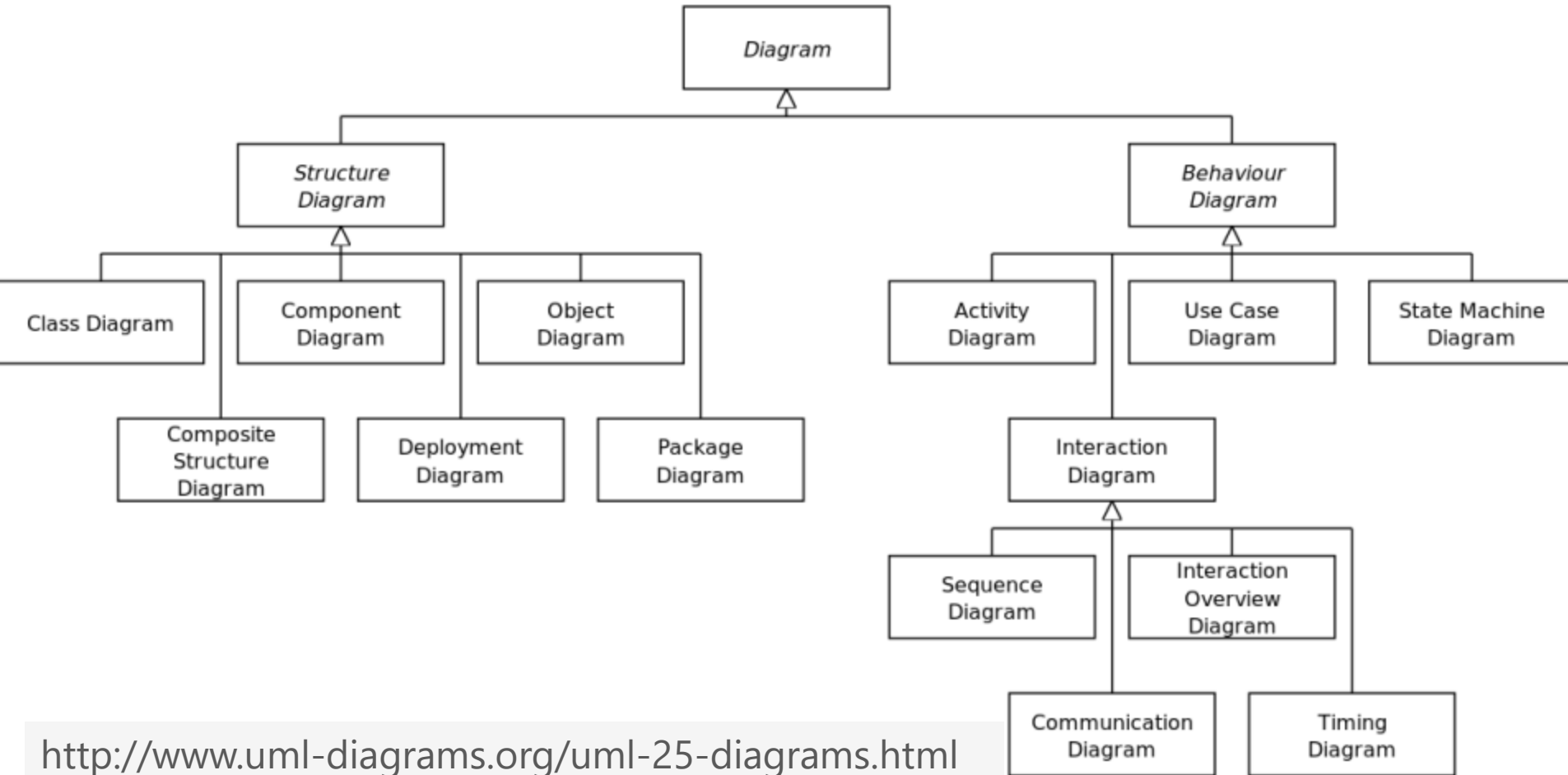
Decomposição funcional de atividades complexas

Visualizar a organização do software em partes e as suas interações

Etc.



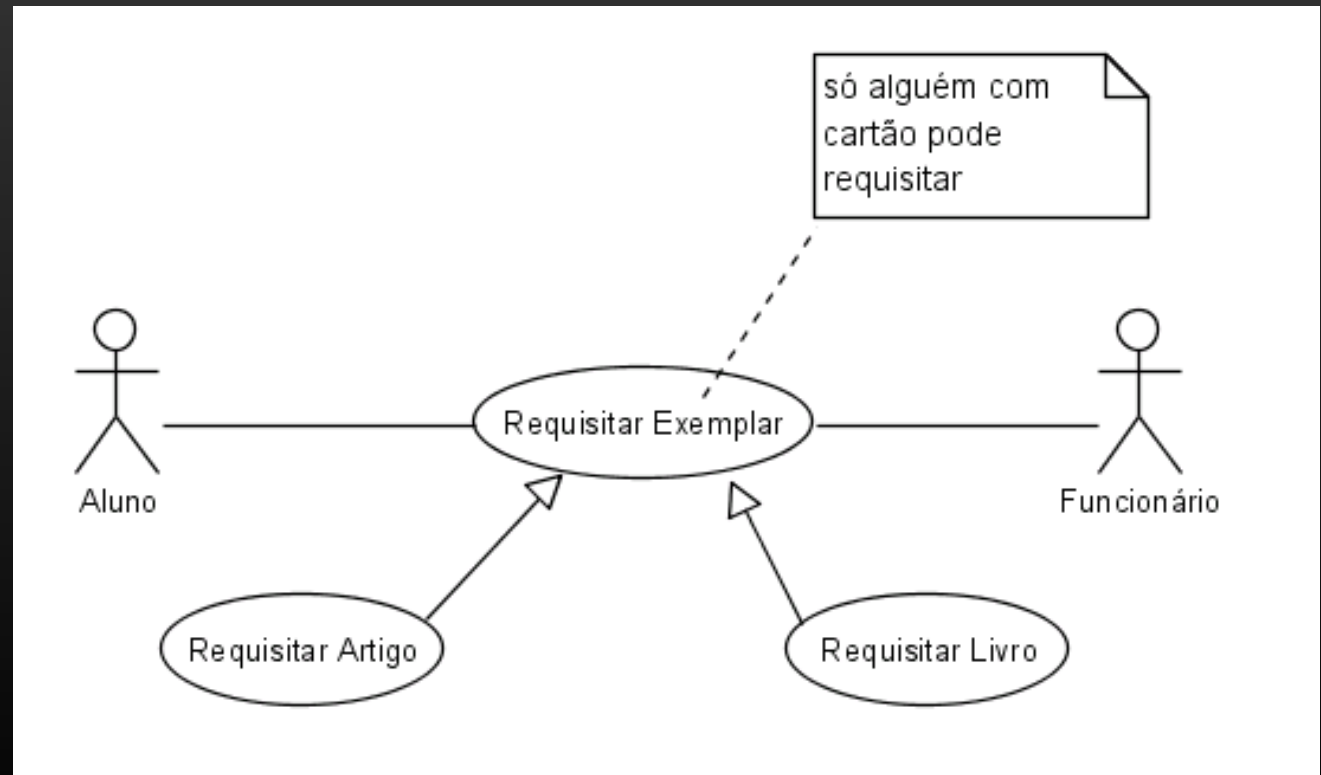
Diagramas da UML 2.x



Elementos comuns

Anotações

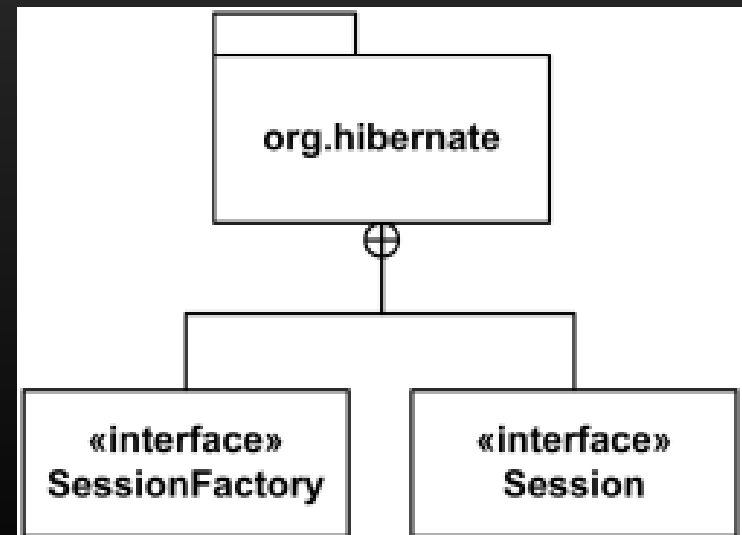
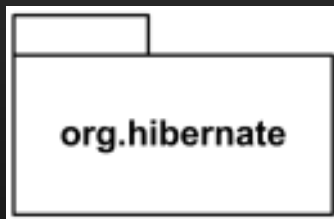
Um comentário que pode ser usado para anotar qualquer elemento



Pacotes

um mecanismo para dividir um modelo em partes

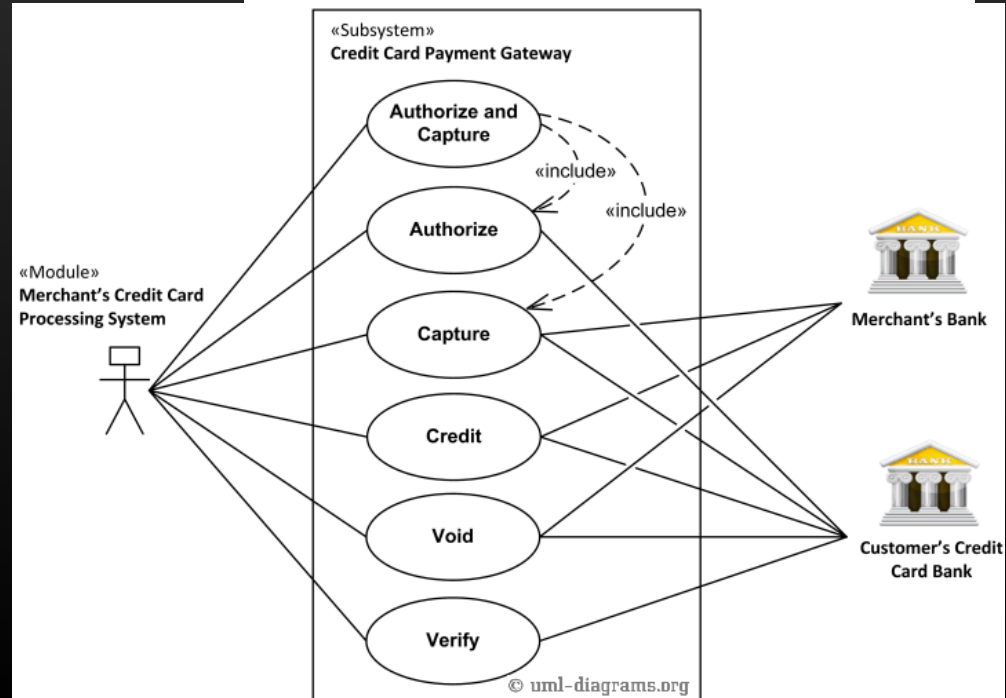
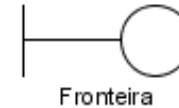
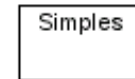
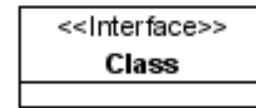
serve como mecanismo genérico para fazer agrupamentos



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)

«Computer»
{Vendor = "Acer",
CPU = "AMD Phenom X4",
Memory = "4 GB DDR2"}
Aspire X1300

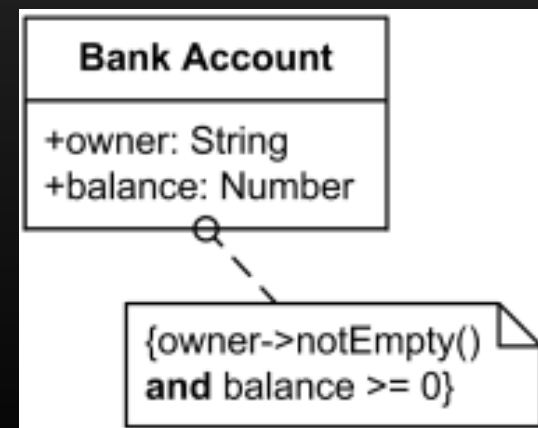
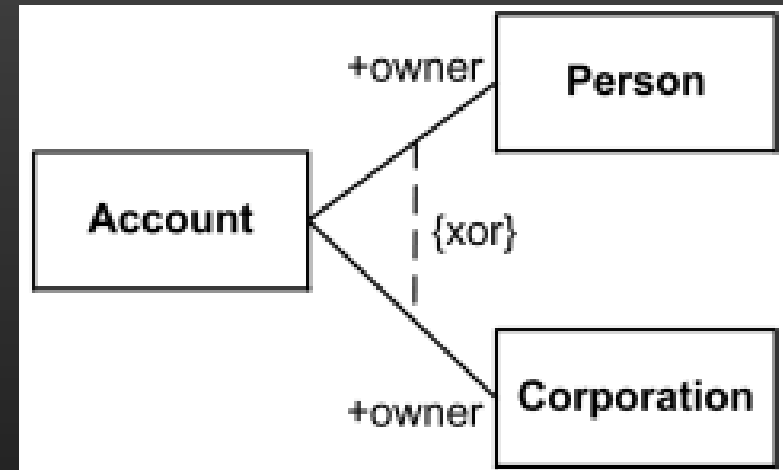
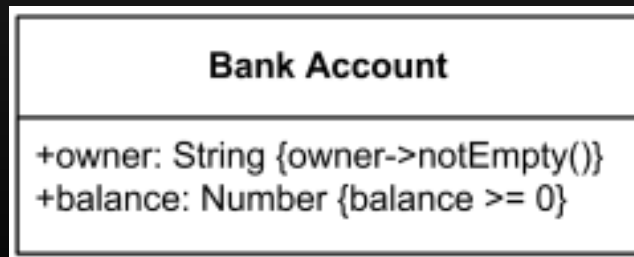
Servidor
{CPUs=3}
admin.exe
inicializa.exe
autoCheck.exe

Restrições

Adicionar regras ao modelo ou
condicionar a sua interpretação

condição ou restrição
relacionada com um ou mais
elementos

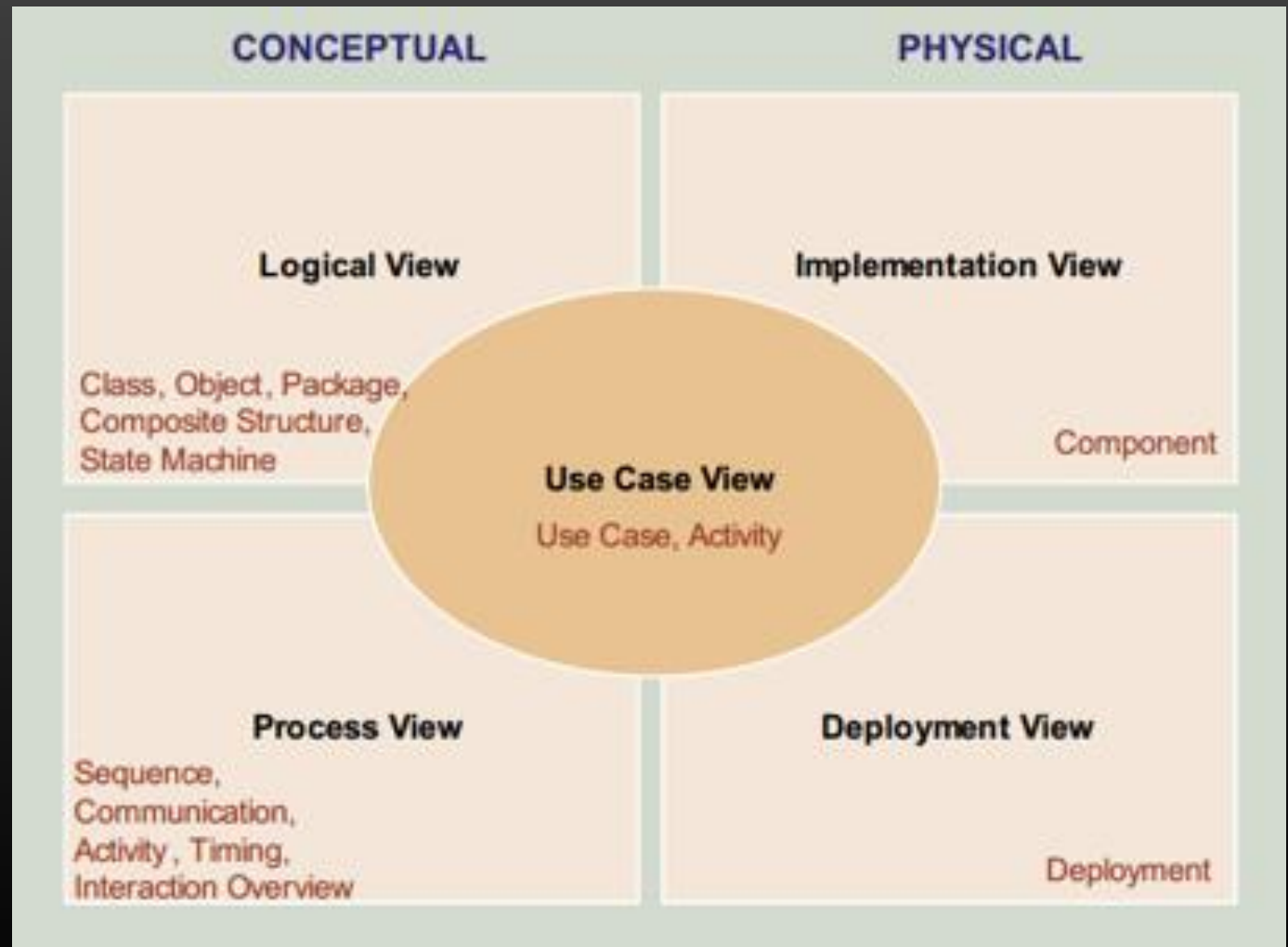
Linguagem própria para declarar
restrições (OCL)



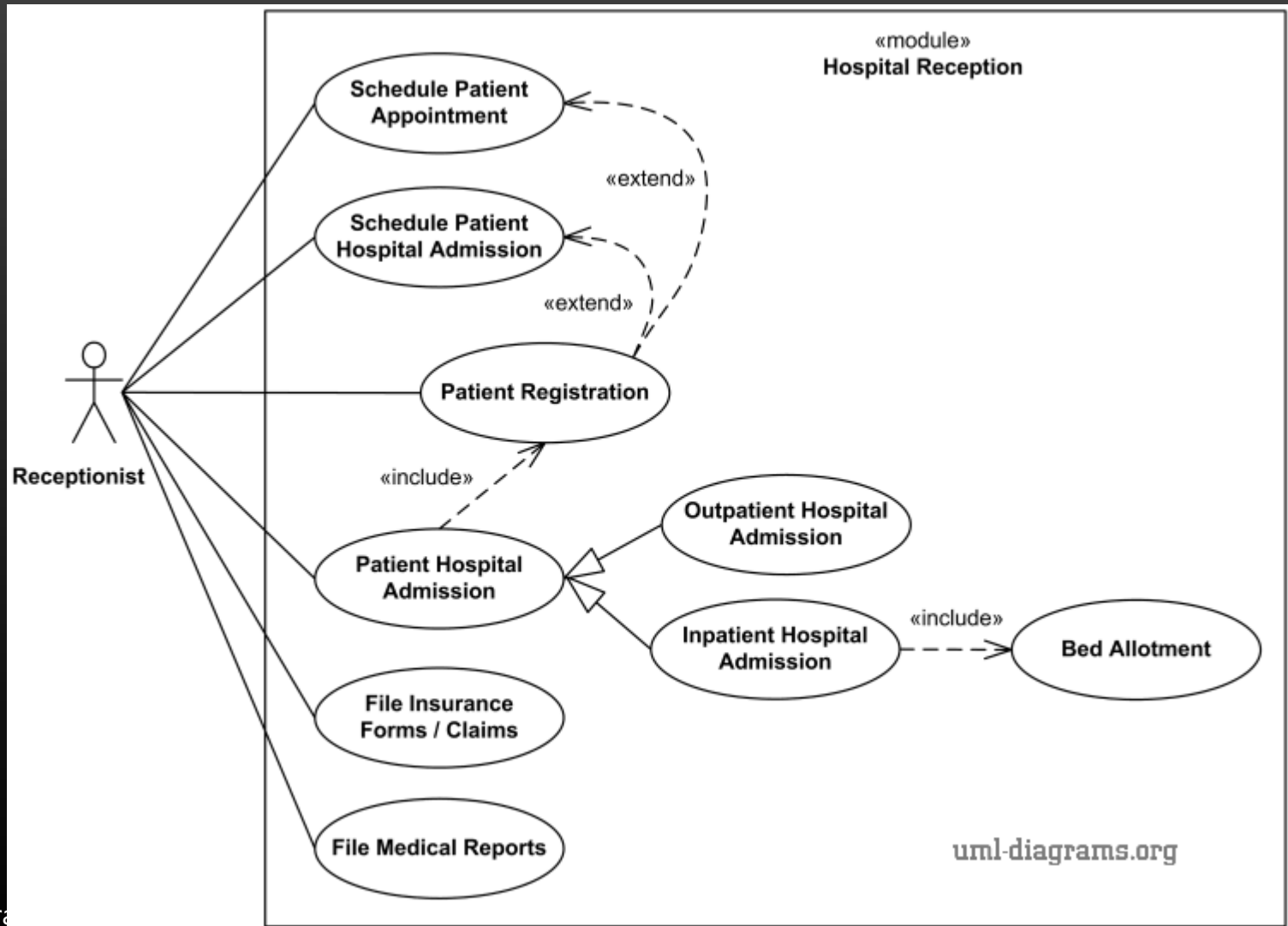
APLICAÇÃO DA UML AO LONGO DO PROCESSO DE DESENVOLVIMENTO

ILÍDIO OLIVEIRA ico@ua.pt
v2017-06-02

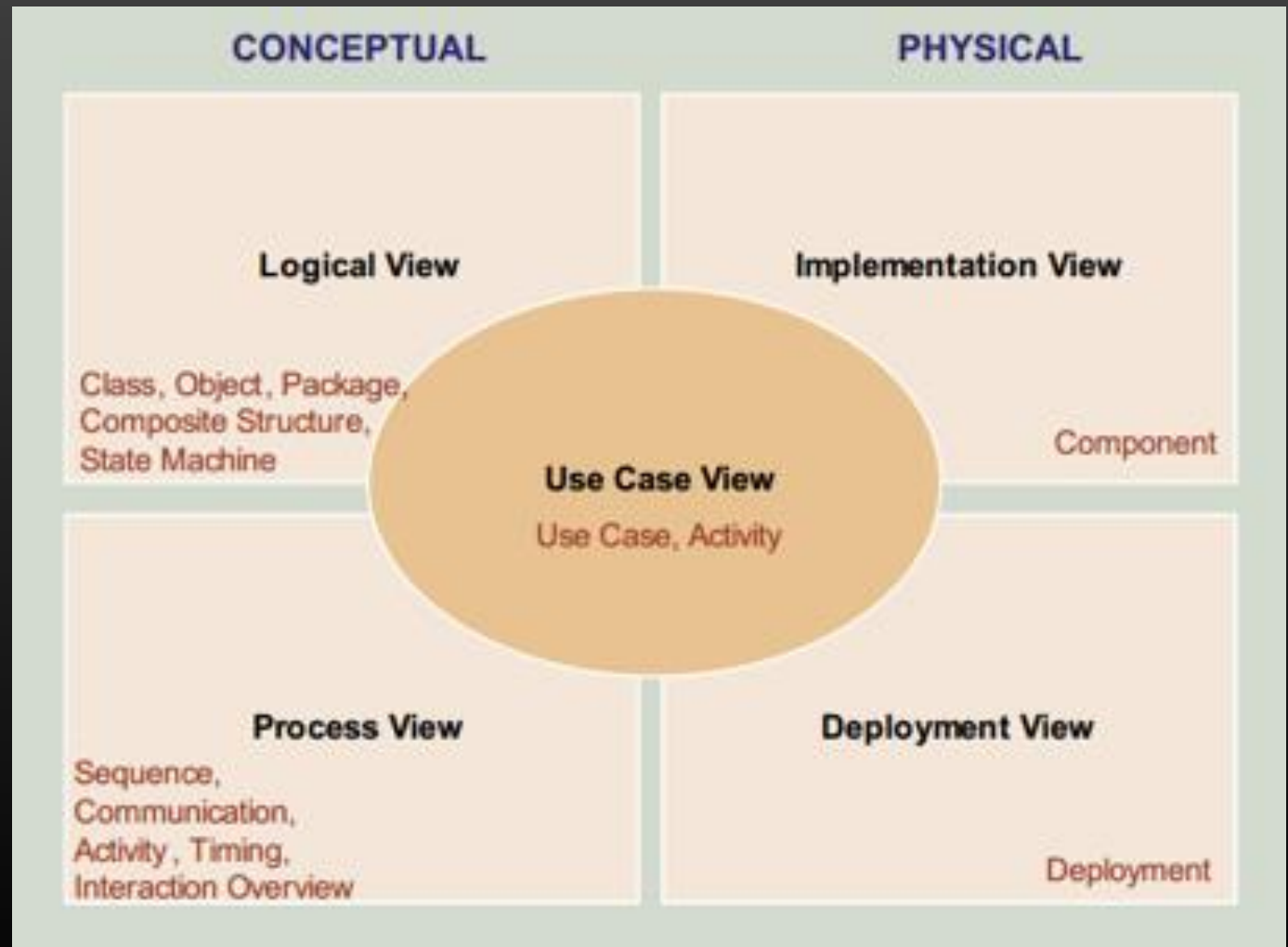
Diversos diagramas para abranger diferentes perspectivas de análise



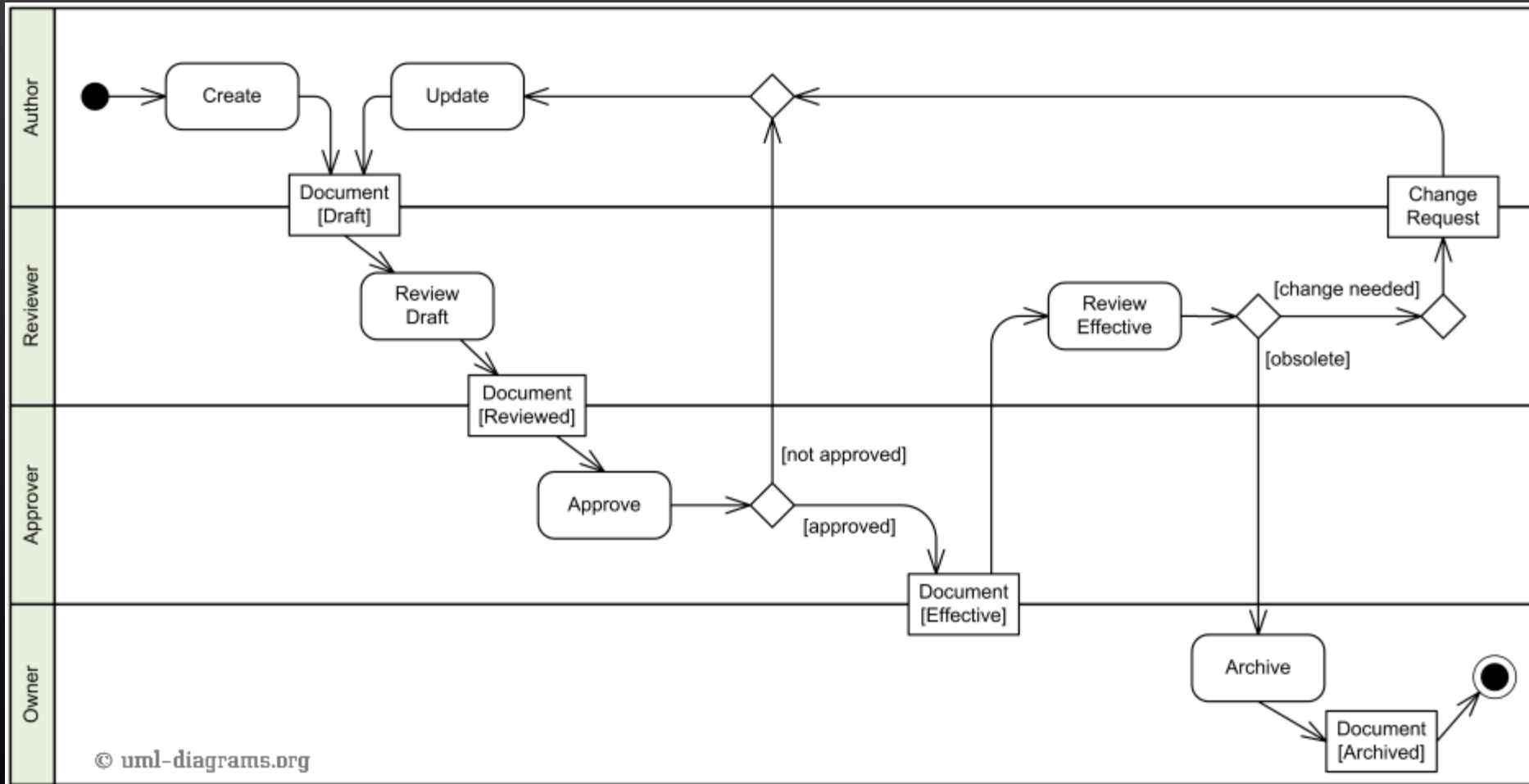
CaU do Sistema: organizar a funcionalidade do sistema em episódios de utilização



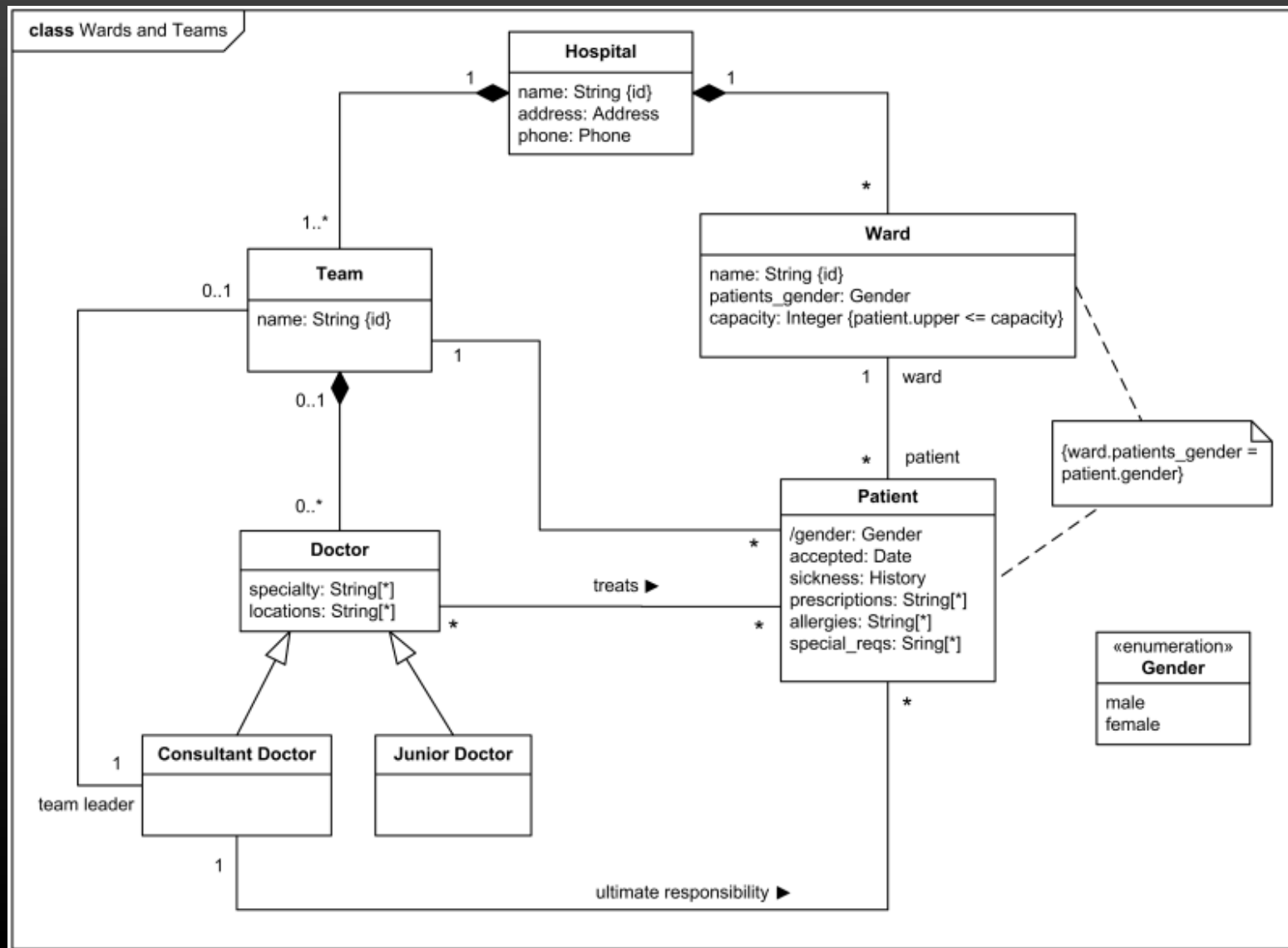
Diversos diagramas para abranger diferentes perspectivas de análise



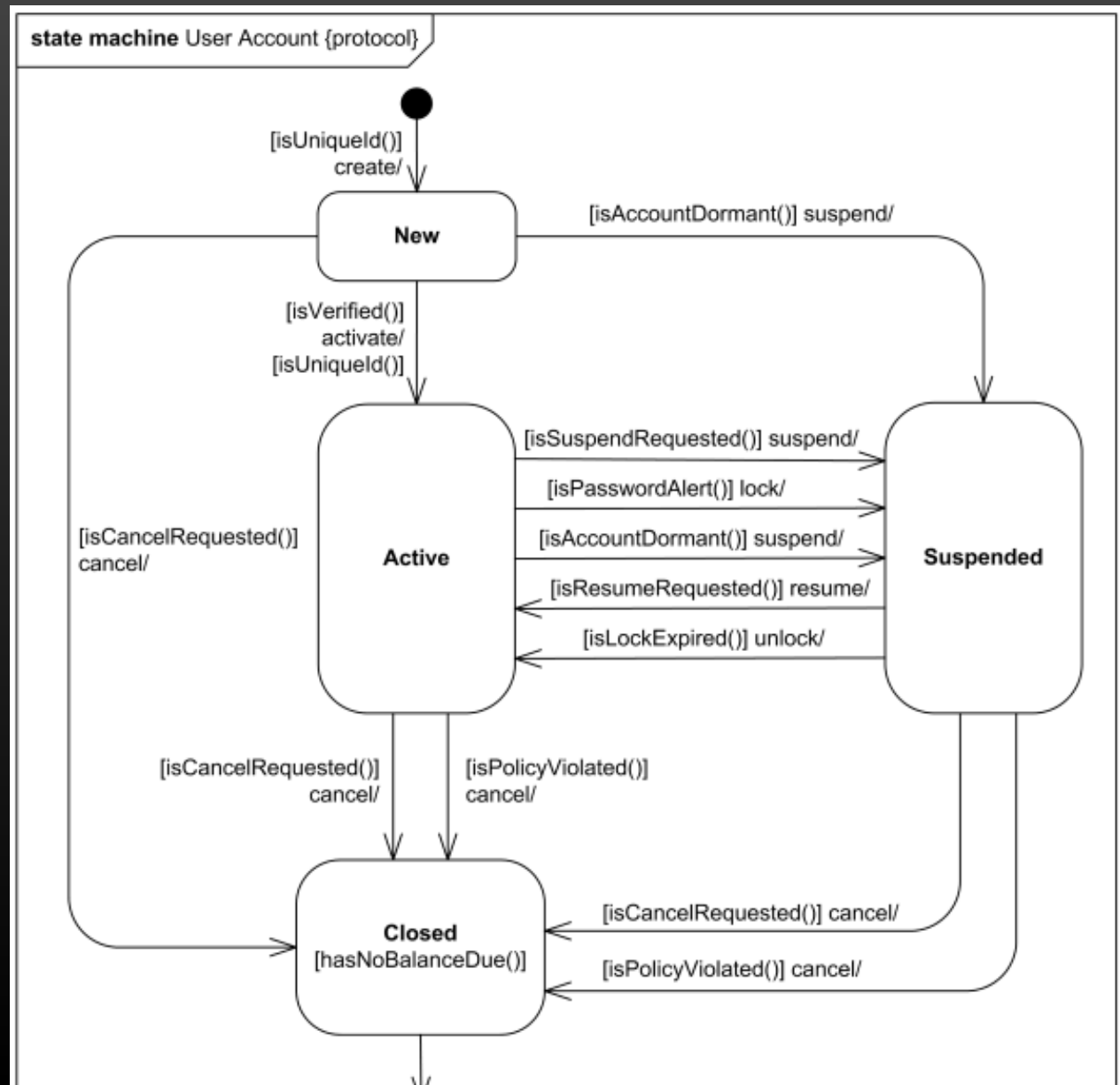
Diagramas de atividades para explicar procedimentos do domínio



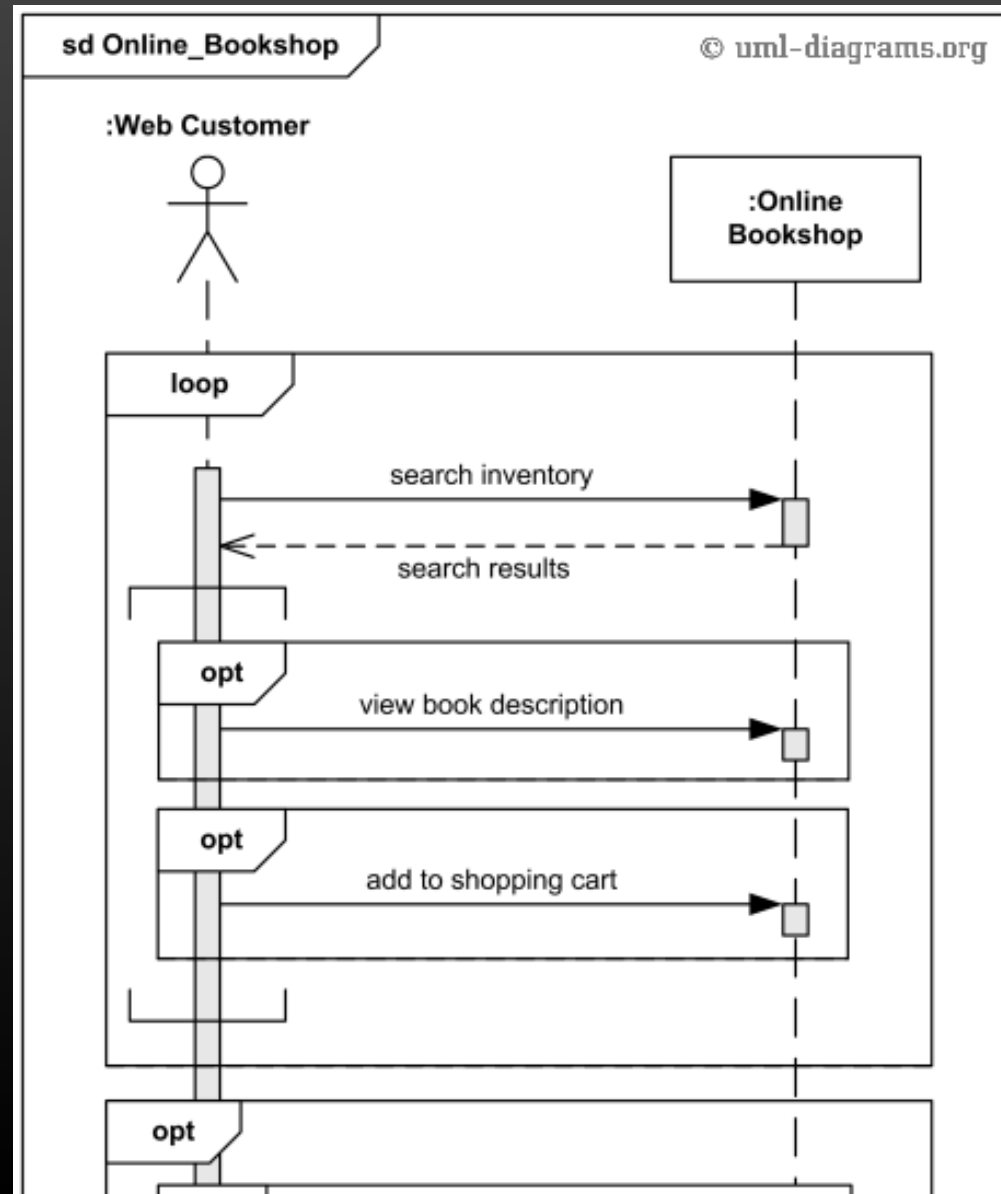
Classes para representar os conceitos da área do problema (modelo do domínio)



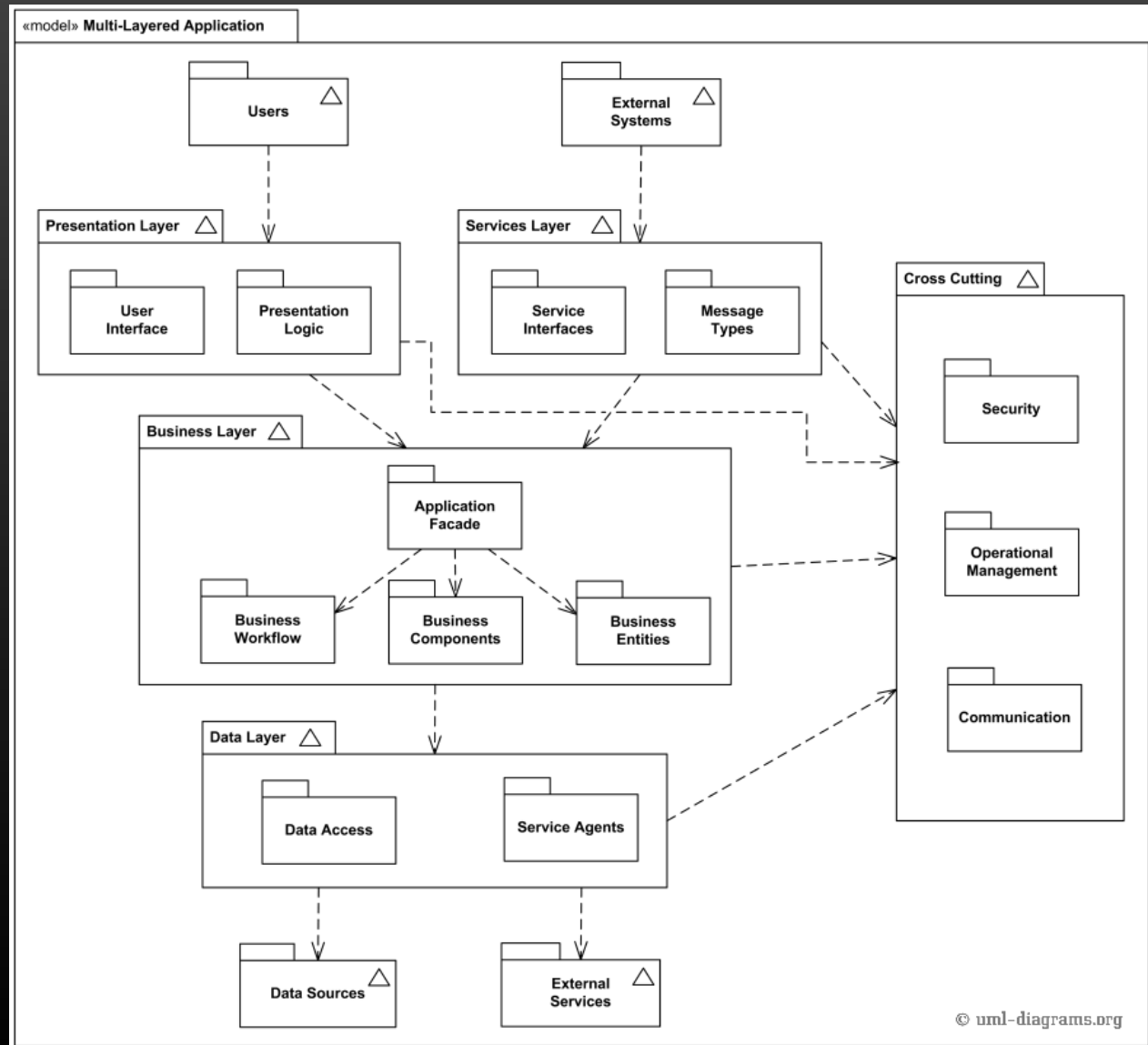
Máquina de estados de entidades/objetos



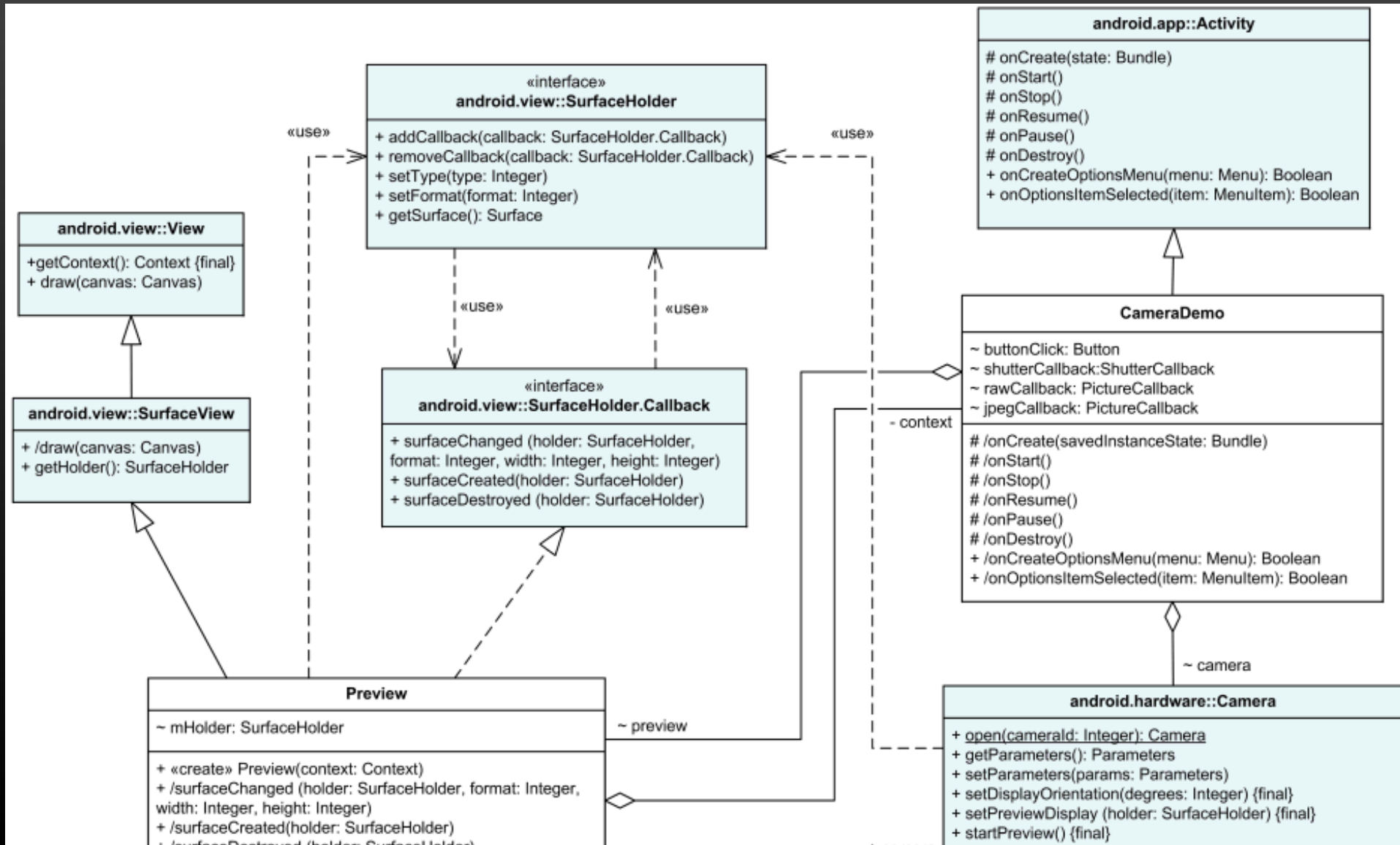
Interação entre atores/objetos



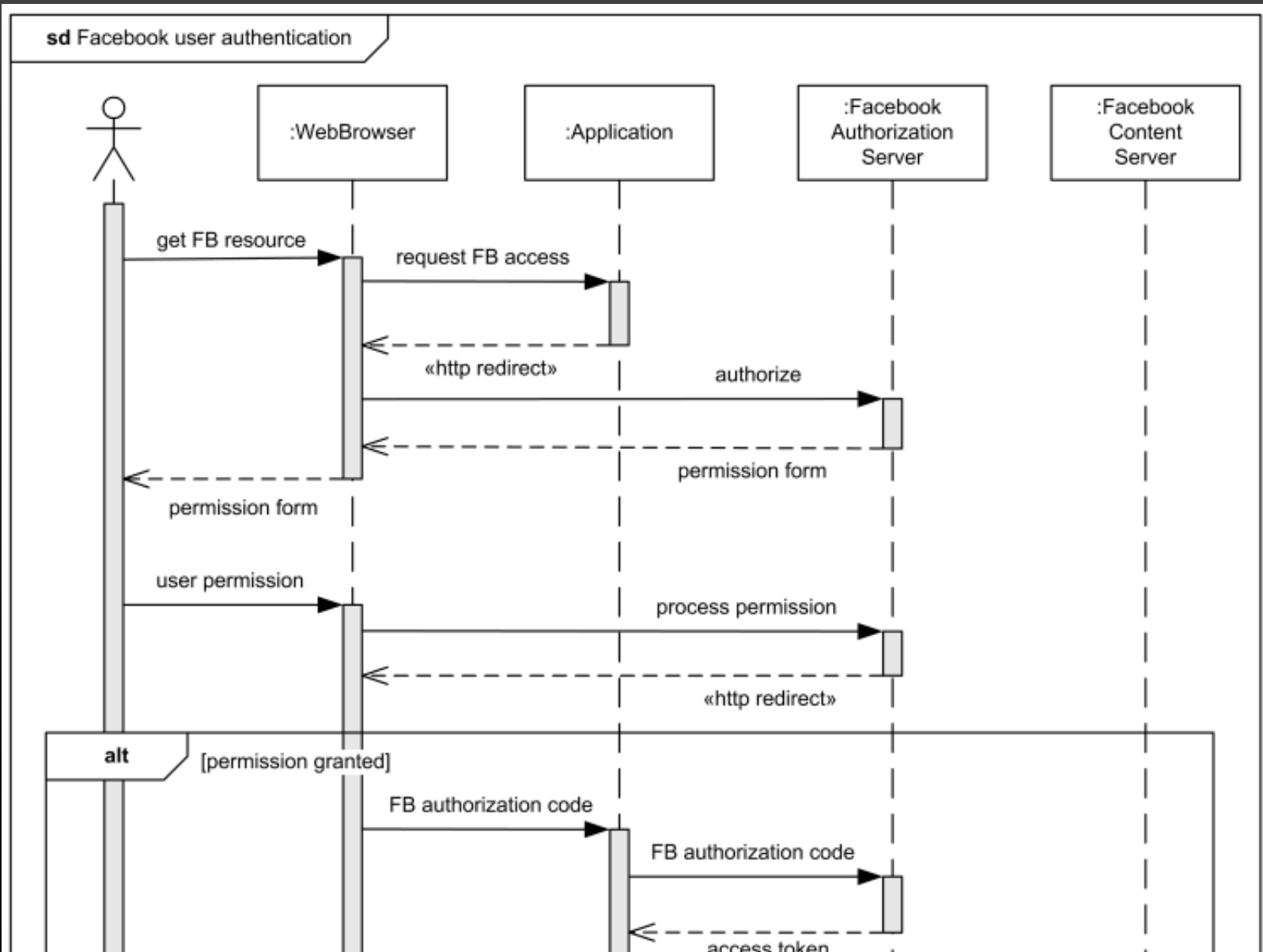
Visualizar a arquitetura lógica com D. Pacotes



Classes para visualizar objetos de um linguagem de programação

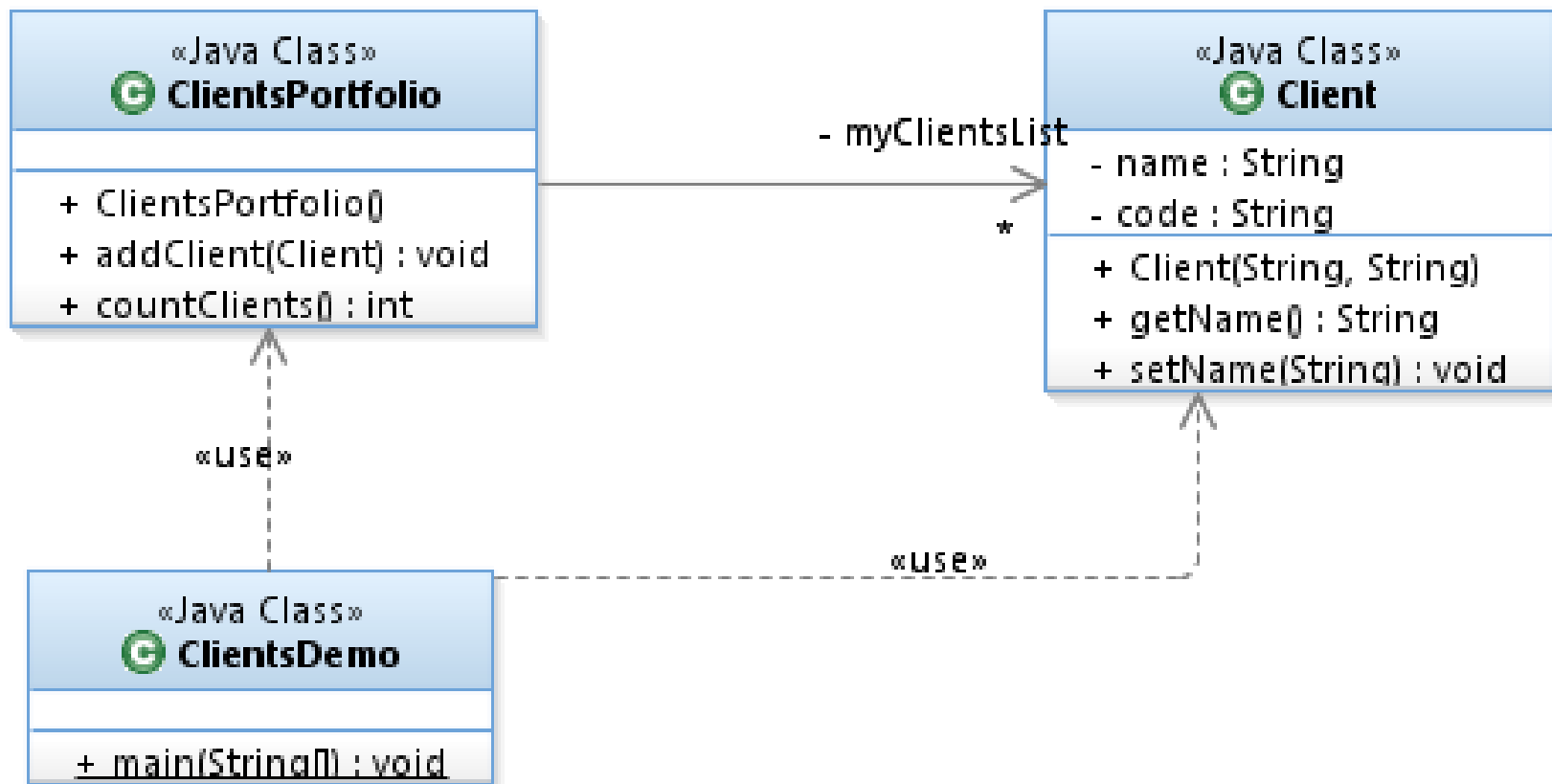


Interações entre componentes do software

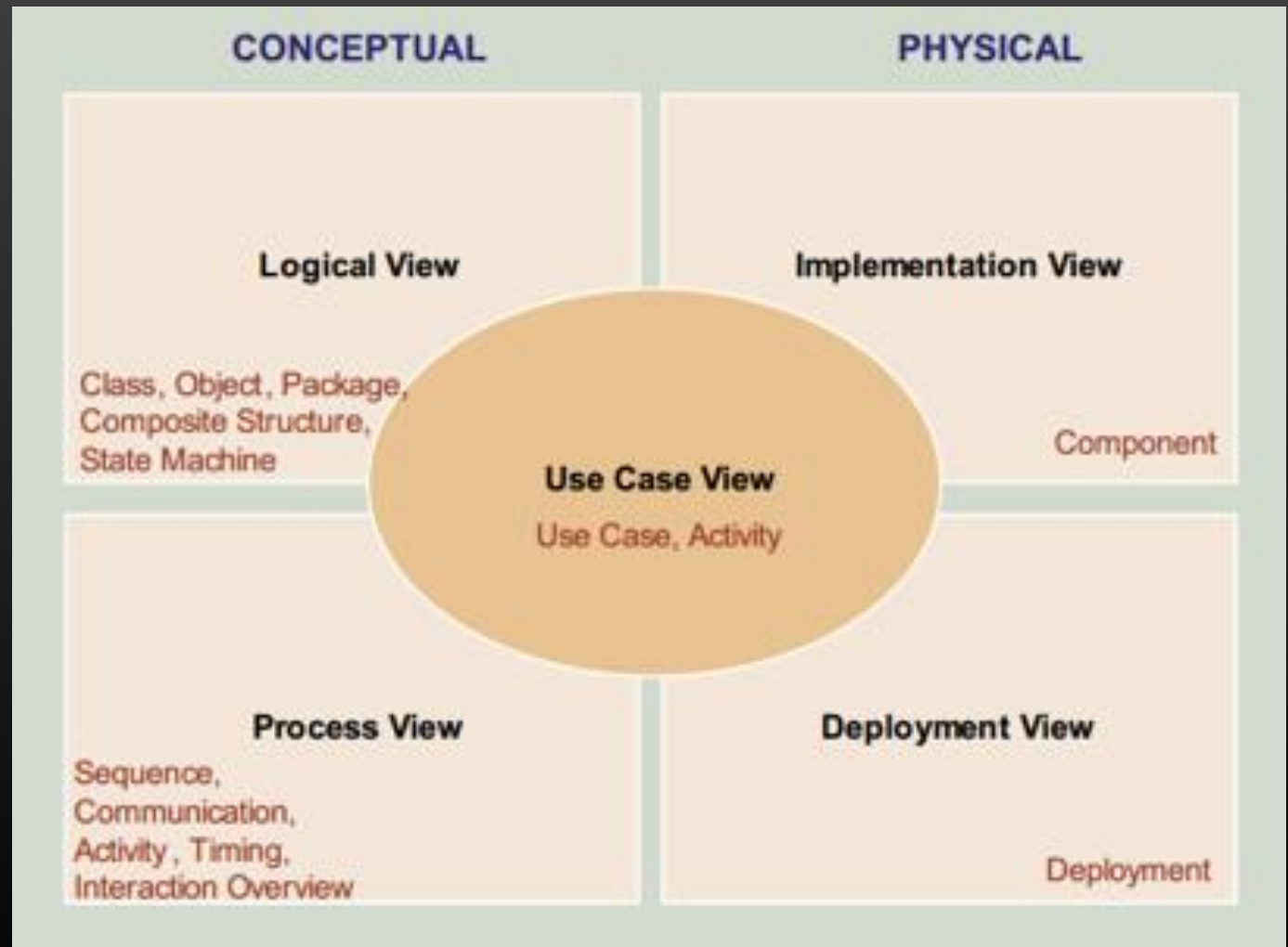


Objetos em código

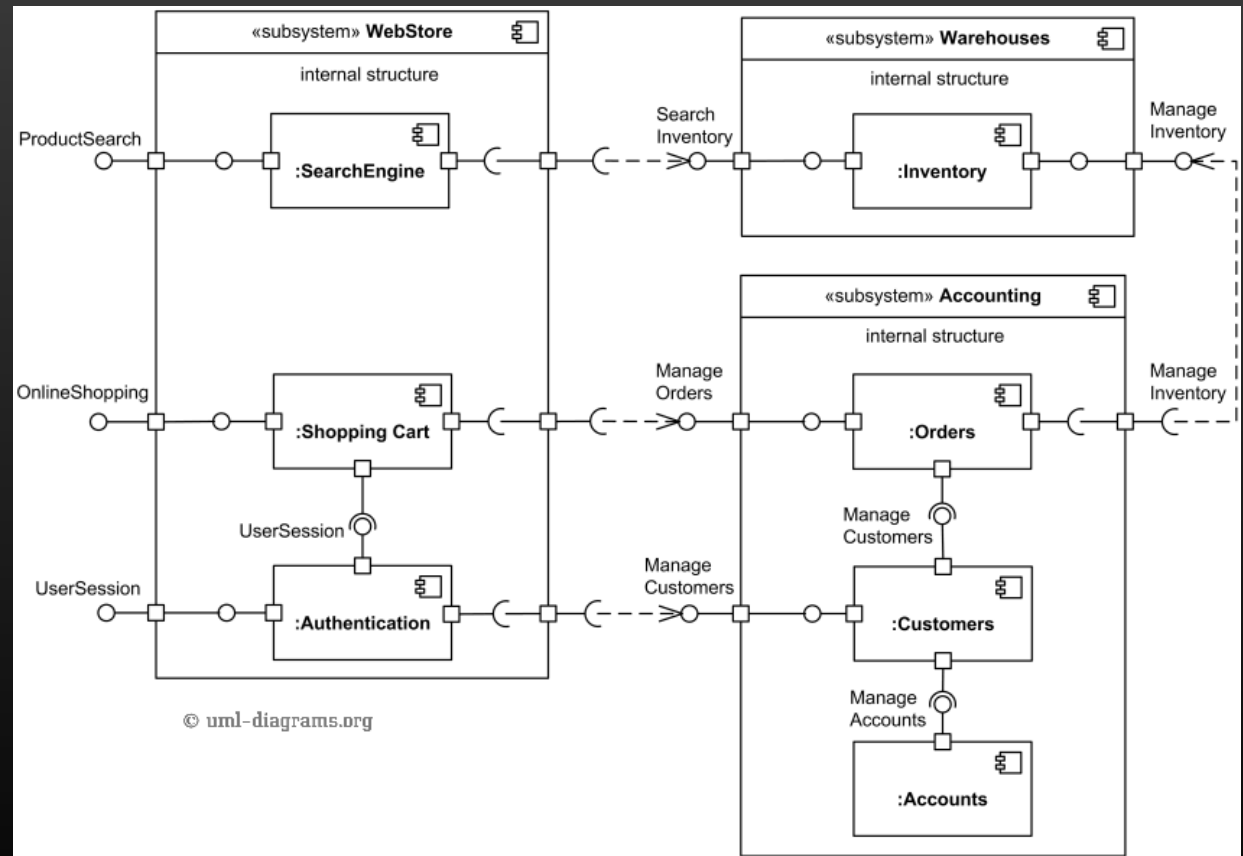
```
public class ClientsPortfolio {  
    private ArrayList<Client> myClientsList;  
  
    public ClientsPortfolio() {  
        myClientsList =new ArrayList<>();  
    }  
    public void addClient(Client newClient) {  
        this.myClientsList.add(newClient);  
    }  
    public int countClients()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



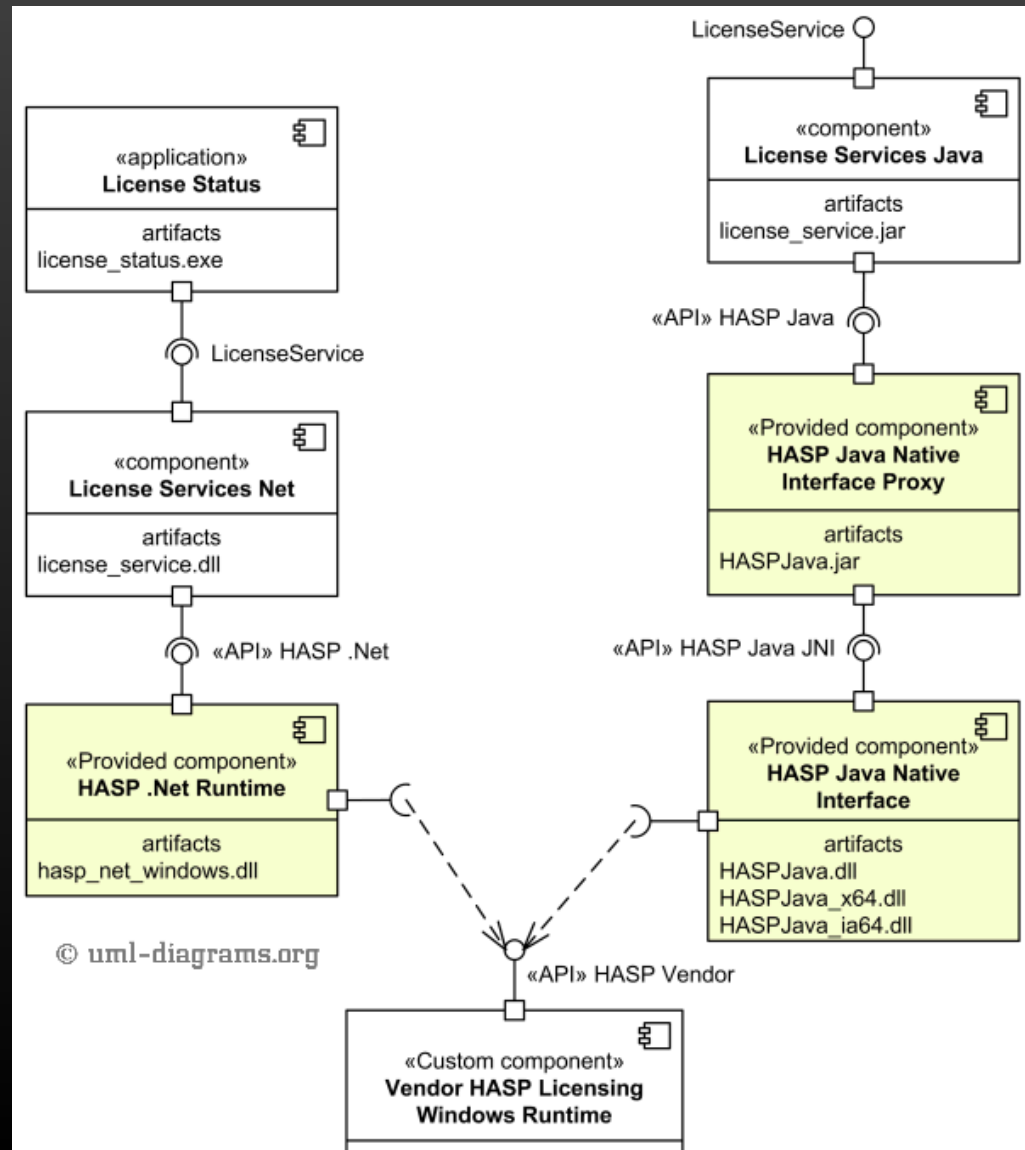
Diversos diagramas para abranger diferentes perspectivas de análise



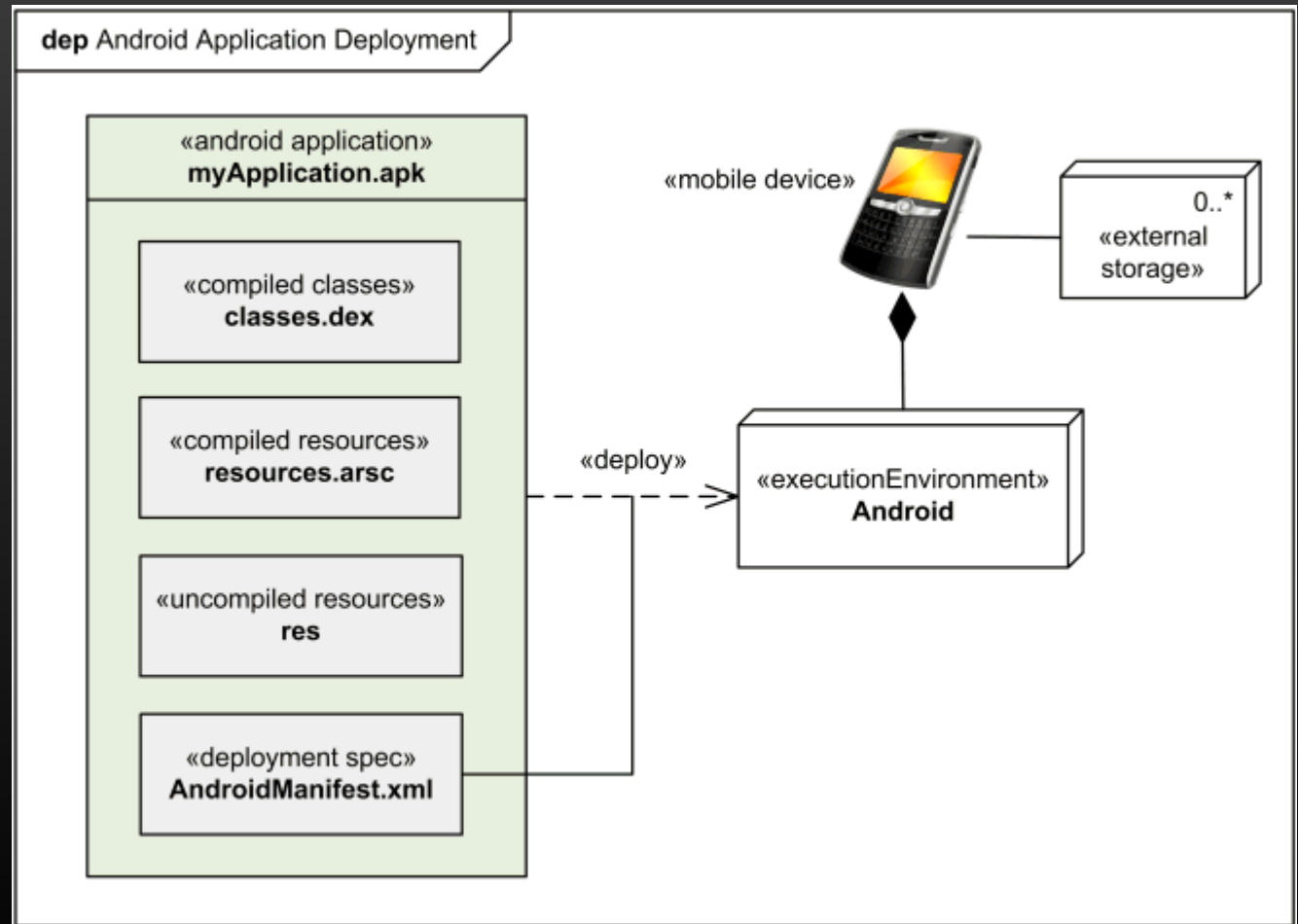
Módulos (executáveis) da solução captados em componentes



Os componentes têm correspondência em artefactos concretos



D. Instalação: mostrar o *setup* para produção



Ferramentas CASE



Readings & references

Core readings	Suggested readings
<ul style="list-style-type: none"><li data-bbox="150 411 691 464">• [Dennis15] – Chap. 1	<p data-bbox="962 411 1767 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 1464 842">→ chap. 10, chap. 15.</p>