

# Métodos y Modelos de Ingeniería de Software INFO 290

Valeria Henríquez N.

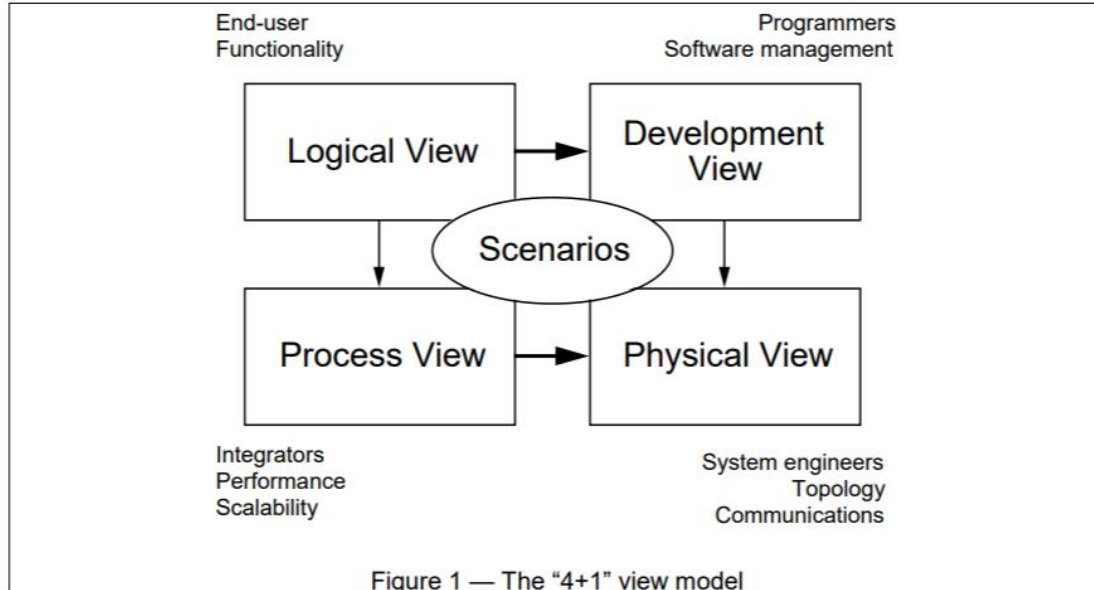
2023



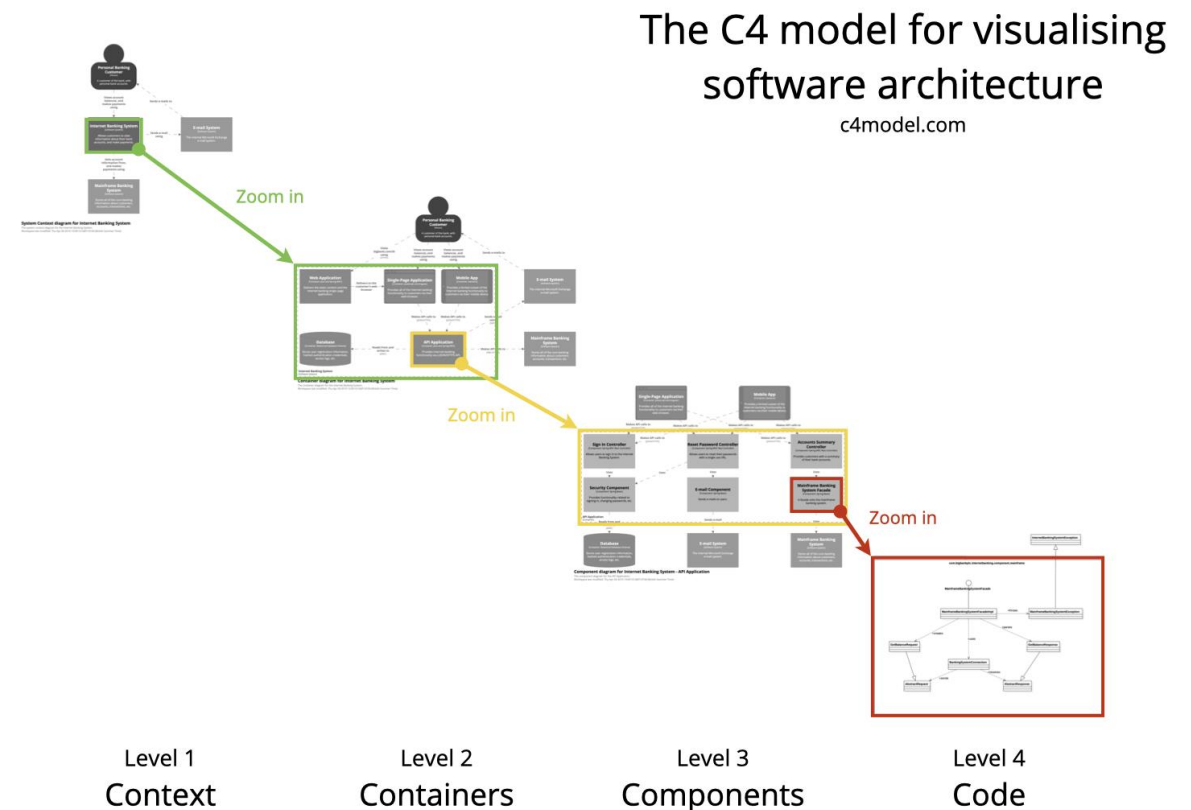
Arquitectura, diseño y  
documentación

# Al menos 2 aproximaciones

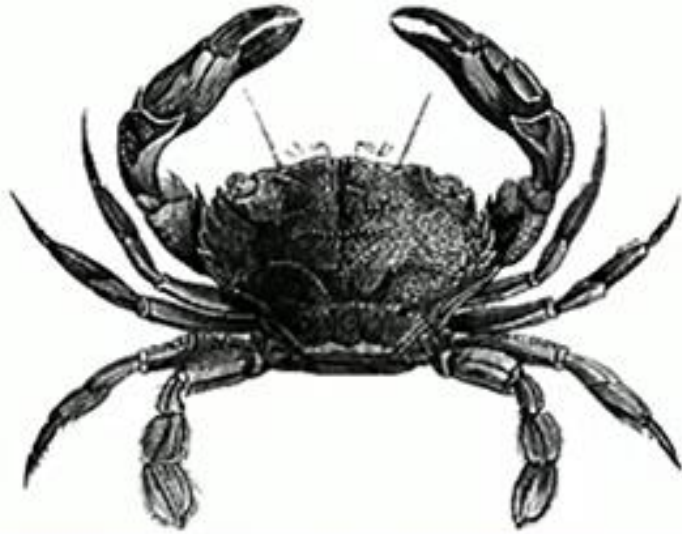
- 4+1 View Model of Software Architecture (Kruchten's Model)



- C4 model (<https://c4model.com/>)



UML



## 97 Ways to Sidestep UML

- #2 "Not everybody else on the team knows it."
- #3 "I'm the only person on the team who knows it."
  - #36 "You'll be seen as old."
  - #37 "You'll be seen as old-fashioned."
  - #66 "The tooling sucks."
  - #80 "It's too detailed."
- #81 "It's a very elaborate waste of time."
  - #92 "It's not expected in agile."
  - #97 "The value is in the conversation."

Mi perspectiva:  
Comunicar abstracciones por sobre  
notación

# UML Diagram Type

## Structural Diagrams

Composite Structure Diagram

Deployment Diagram

Package Diagram

Profile Diagram

**Class Diagram**

Object Diagram

Component Diagram

## Behavioral Diagrams

Activity Diagram

Use Case Diagram

State Machine Diagram

Interaction Diagram

Sequence Diagram

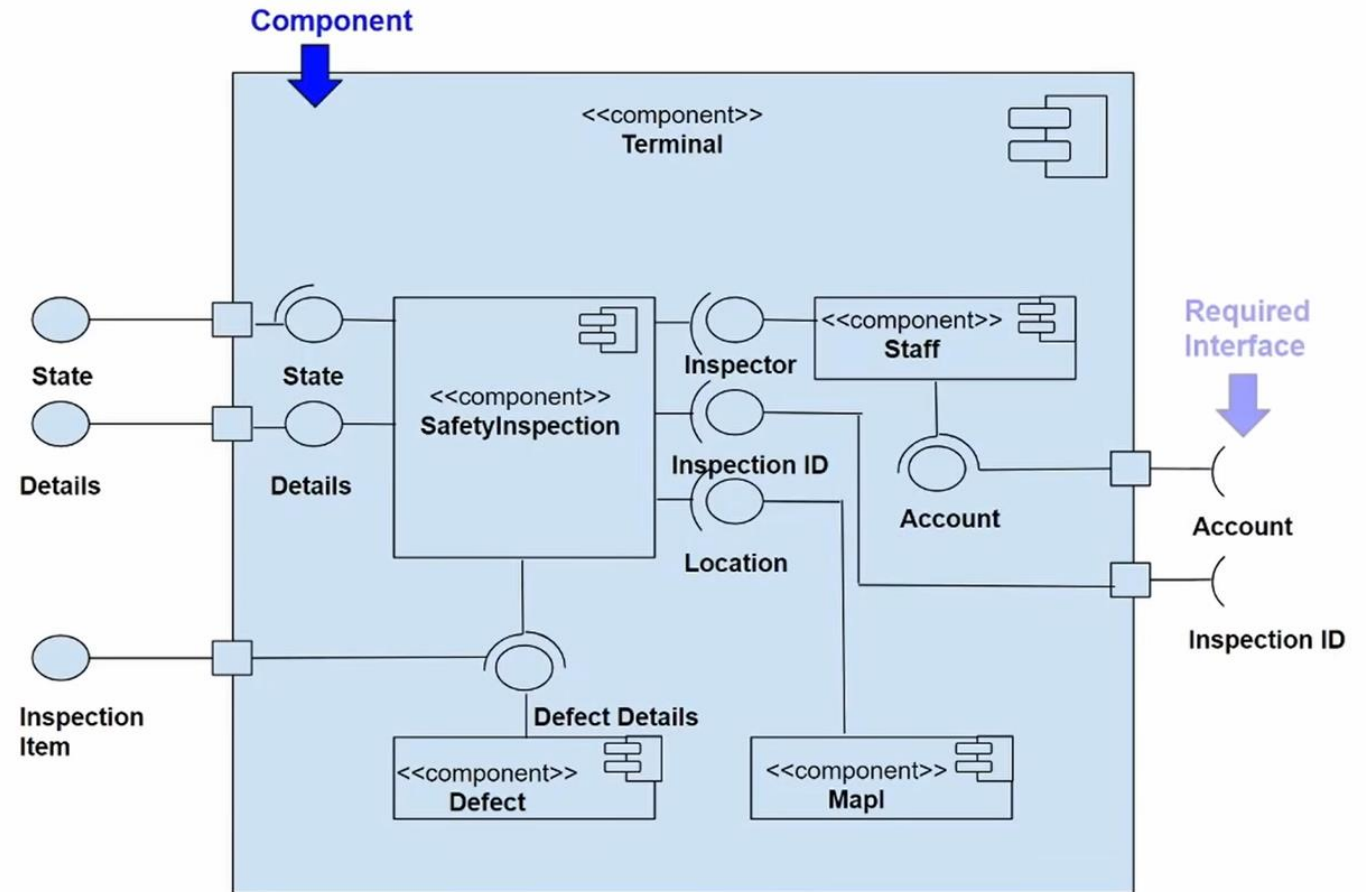
Communication Diagram

Interaction Overview Diagram

Timing Diagram



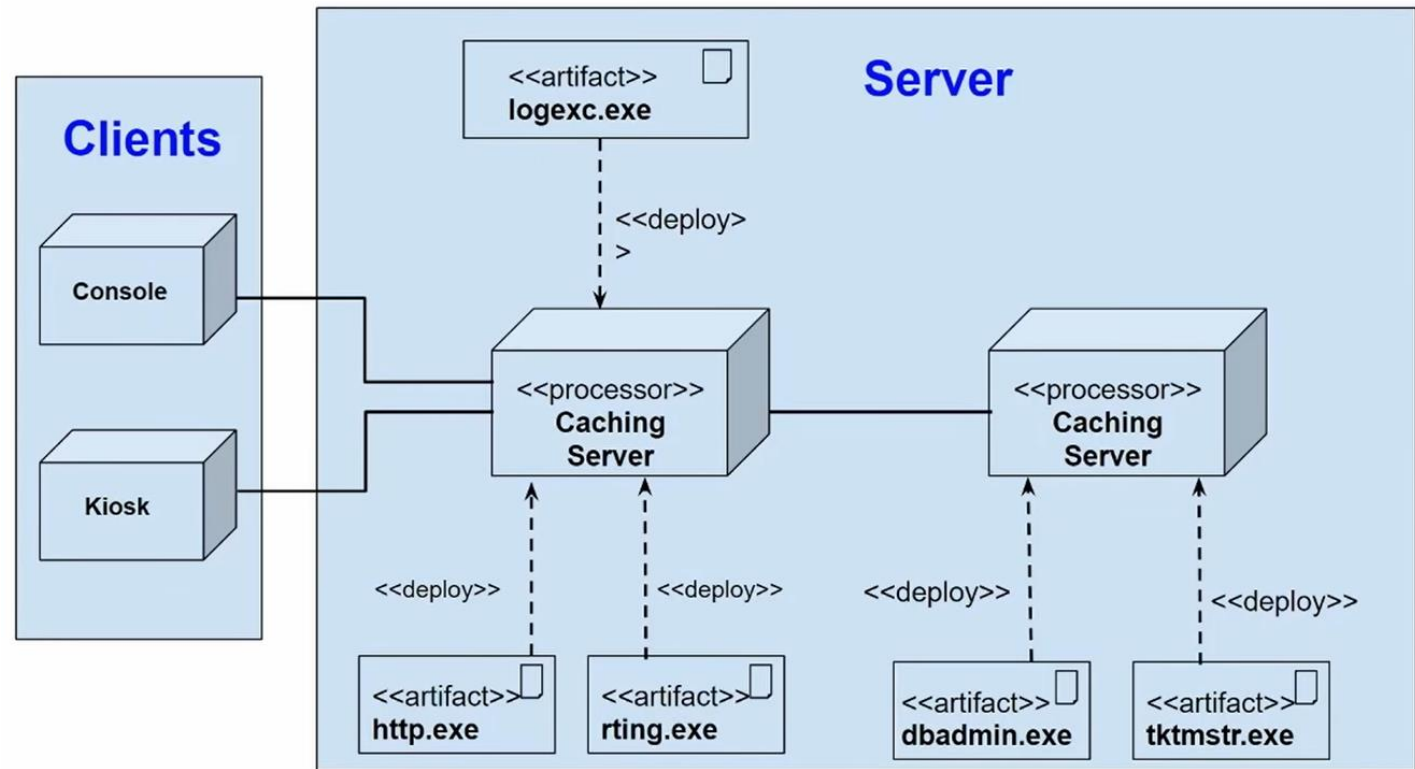
# Component diagram



- Visualizar cómo interactúan las piezas de un sistema y qué relaciones tienen entre ellas
- Se centran en la estructura de alto nivel del sistema y no en los detalles como atributos y métodos
- Son una vista estática del sistema de software y representan el diseño del sistema
- Pueden diagramarse diferentes niveles y representar sistemas enteros y también subsistemas

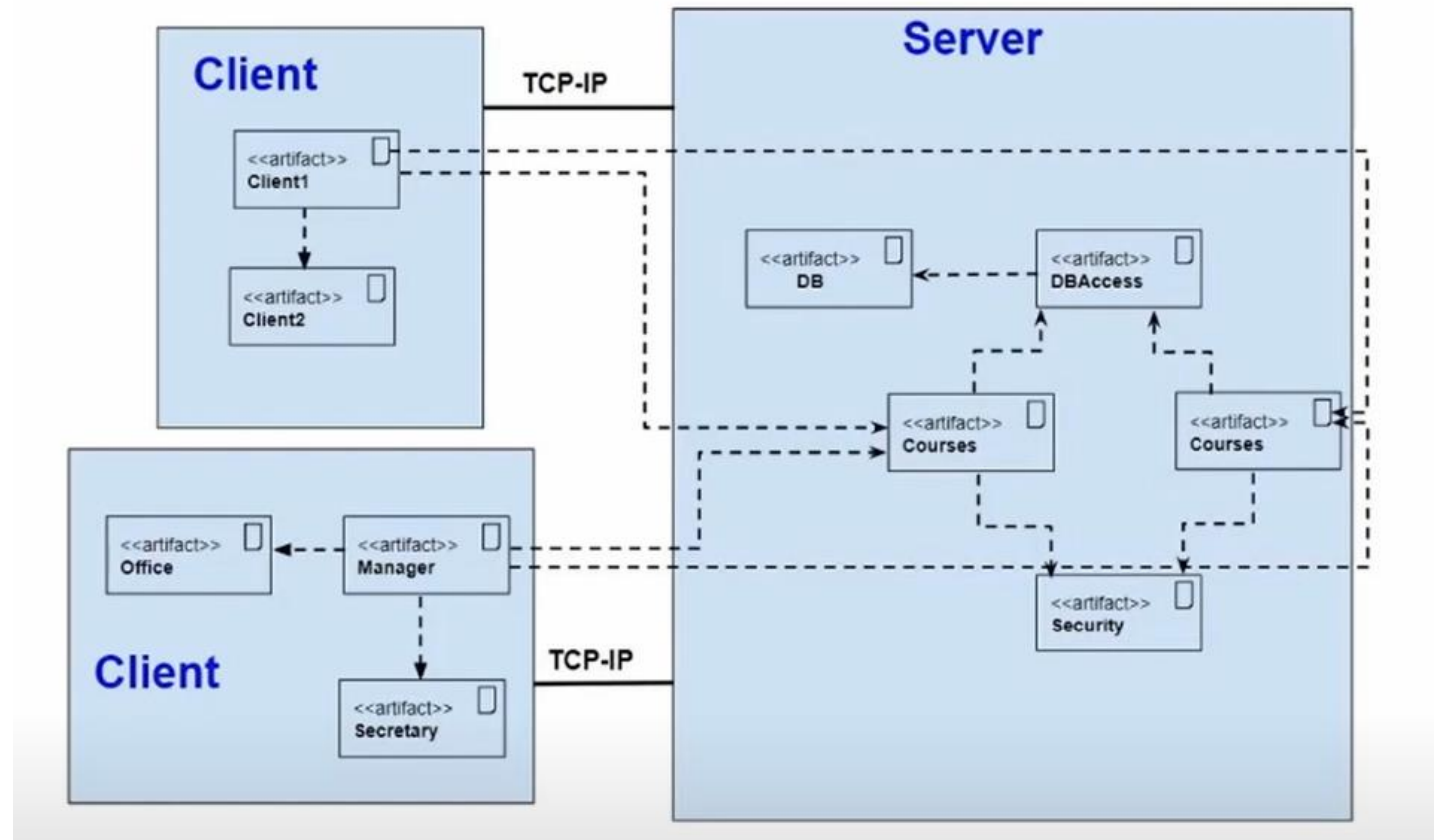


# Deployment diagram

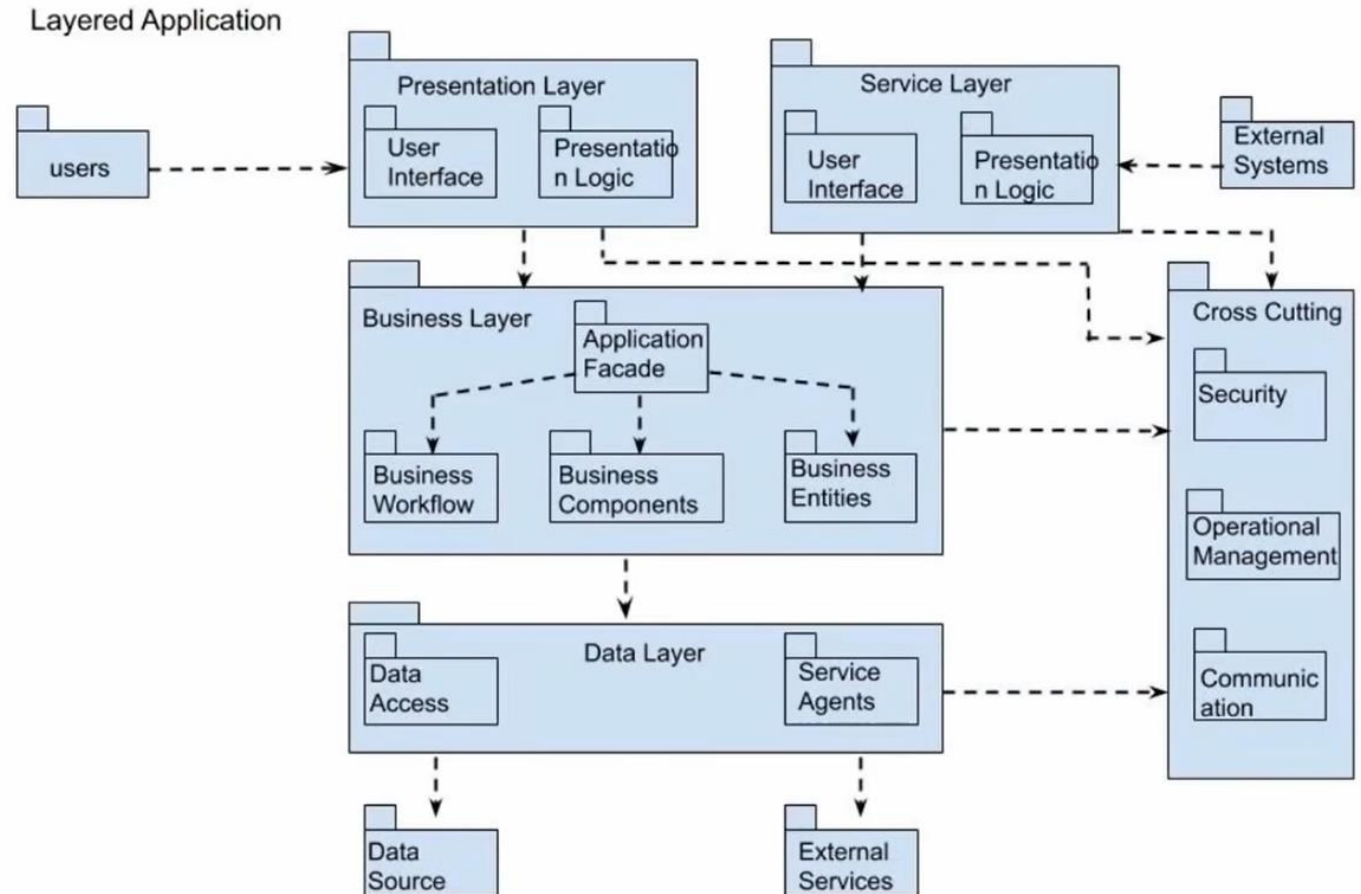


- visualizar los detalles de implementación, incluyendo los artefactos, ejecutables, archivos de configuración, etc.
- A nivel de especificación, que ofrece una visión general de los artefactos y los objetivos de implementación
- A nivel de instancia, que ofrece una visión específica, identificando máquinas y dispositivos de hardware específicos
- Es importante completar la jerarquía de objetivos de implementación, empezando desde el nivel más alto de información , como el nombre de la aplicación hasta el dispositivo y el sistema operativo

# Deployment diagram



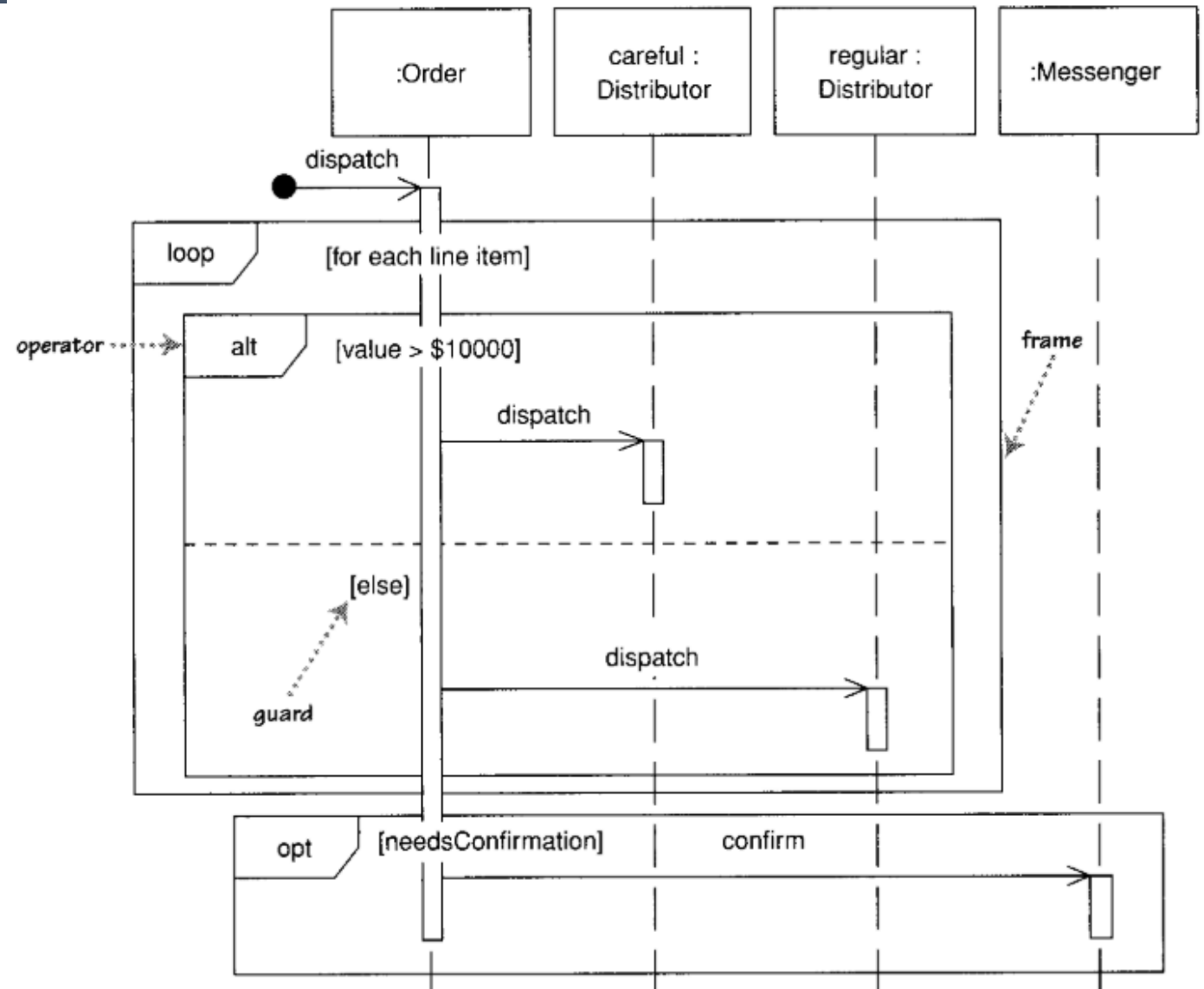
# Package diagram



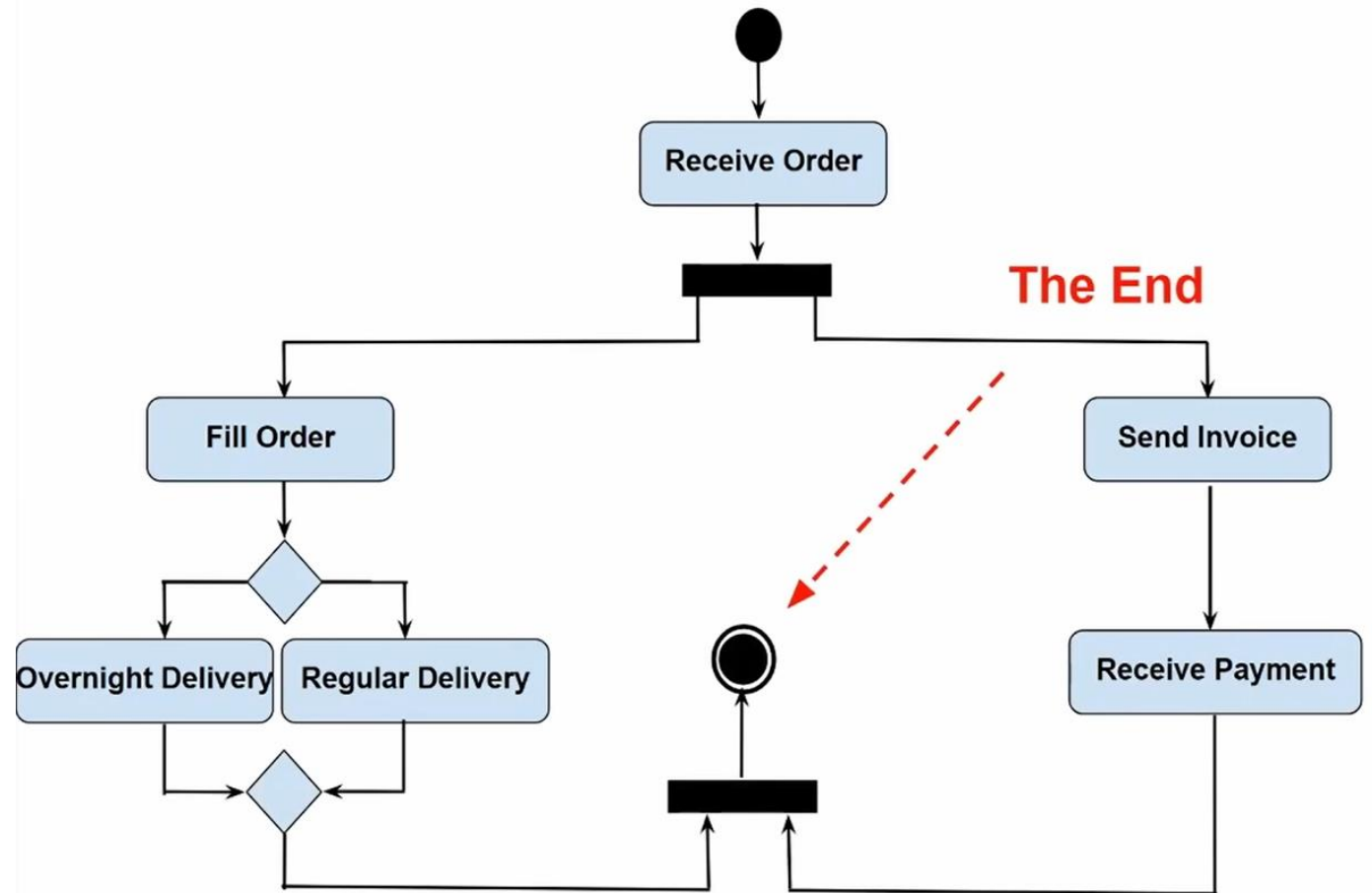
- Es más flexible que los otros y muestra un alto nivel de agrupaciones
- Mostrar cómo los paquetes y sus elementos están relacionados y organizados en un sistema de software
- se representan como carpetas con pestañas y contienen elementos relacionados

# Diagramas de comportamiento

# Sequence diagram

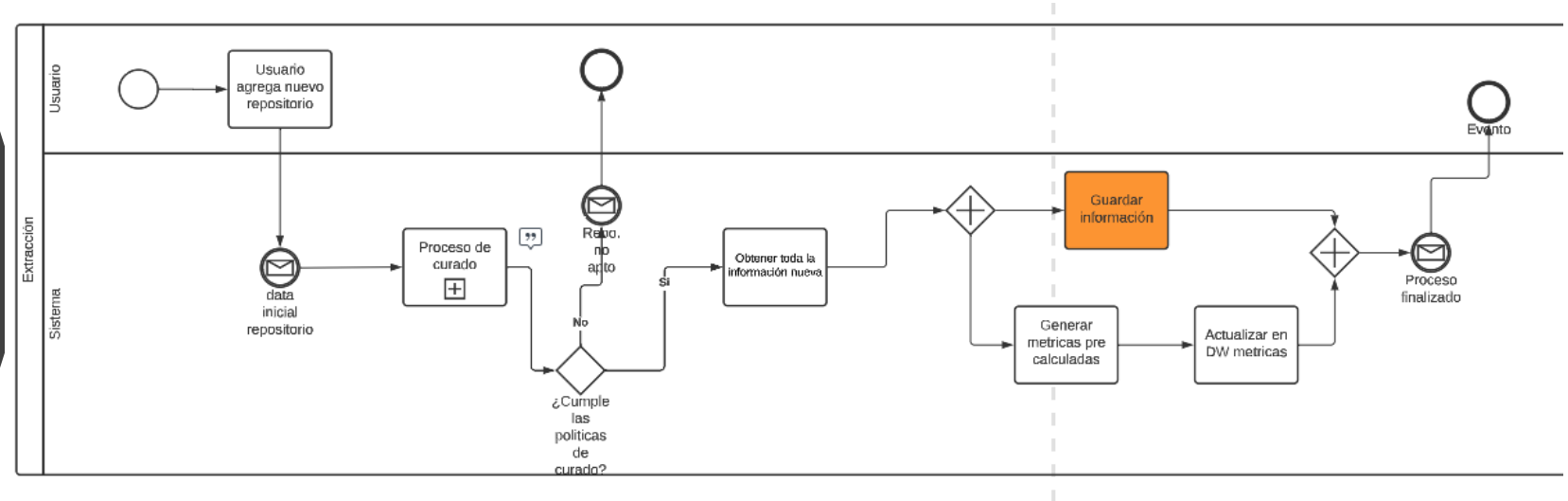


# Activity diagram



- capturar el comportamiento dinámico del sistema,
- permite mapear el comportamiento de alto nivel de un sistema
- permite mapear la ramificación en flujos alternativos

# BPNM





Respecto a su proyecto elegido :

1. ¿Qué diagramas UML realizó?
2. ¿Cuáles de ellos sirven para comunicar la arquitectura?
3. **Mejorar/Crear 2 diagramas (uno de estructura y uno de comportamiento)**

Use <https://lucid.app/> para crear sus diagramas

# Tarea

- Lista de videos 4

<https://drive.google.com/drive/u/2/folders/13Arjx015yWFvpVCd2R7jvPfHPraUGLMN>

