

**UPN, PASIÓN POR  
TRANSFORMAR VIDAS**

**UNIDAD 2:  
DISEÑO ARQUITECTÓNICO**

**SESIÓN 5:  
Arquitectura de Software**

[eric.coronel@upn.pe](mailto:eric.coronel@upn.pe)

# UNIDAD 2: DISEÑO ARQUITECTÓNICO



## SESIÓN 5: Arquitectura de Software





**EL ÉXITO ES LA SUMA DE  
PEQUEÑOS ESFUERZOS,  
REPETIDOS DÍA TRAS DÍA.**

Robert Collier



1. ¿Qué diferencia existe entre un caso de uso y un requisito funcional en el desarrollo de software?
2. ¿Cuáles son los principales componentes de un diagrama de clases UML y qué representan?
3. ¿Qué entiendes por patrones arquitectónicos en el diseño de software y menciona dos ejemplos?

# REFLEXIONA



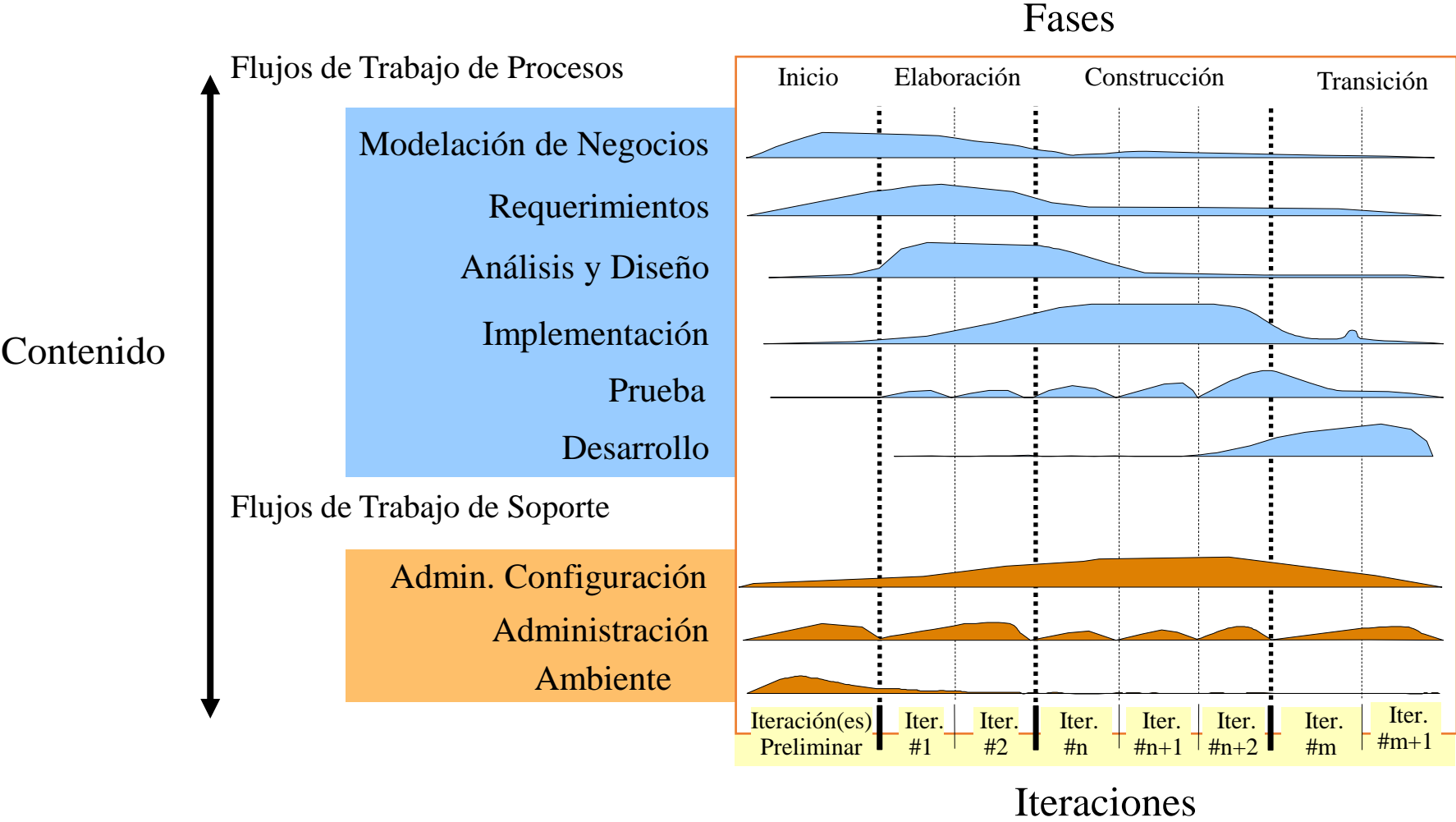


Al finalizar la unidad, el estudiante diseña la arquitectura de un producto de software, haciendo uso de patrones arquitectónicos, patrones grasp, elementos del modelo de diseño, clases de diseño, subsistemas, interfaces, herramientas CASE y consideraciones; cumpliendo con los requerimientos fundamentales del producto.



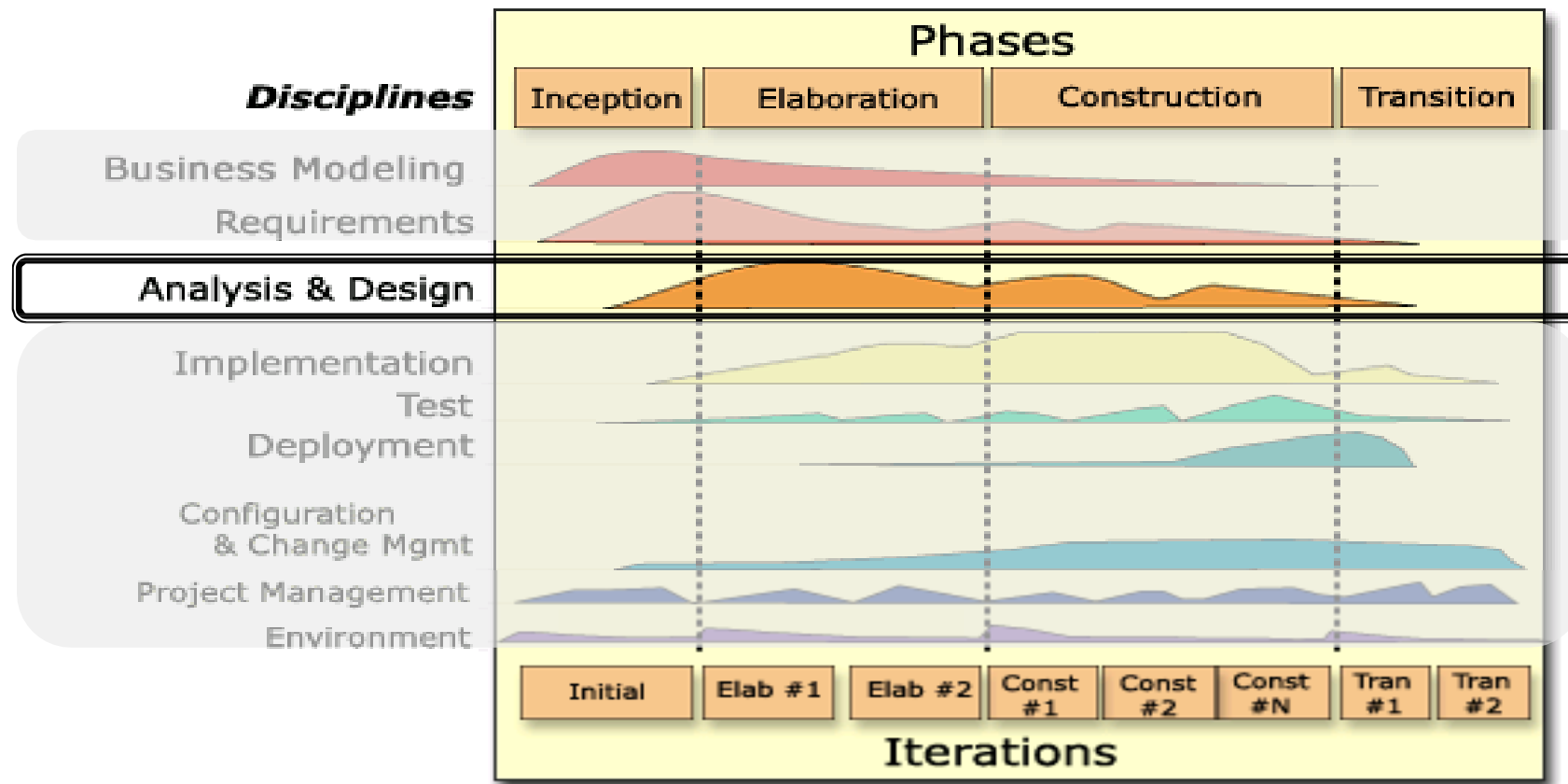
Al finalizar la sesión, el estudiante será capaz de aplicar patrones GRASP en el modelado de análisis de software, creando diagramas de clases e interacción, desarrollando modelos de datos. Este logro contribuye al desarrollo de competencias específicas relacionadas con el diseño arquitectónico y el uso de herramientas CASE en el proceso de análisis y diseño de software.

# RUP: SOPORTE TEÓRICO





# I. ¿EN QUÉ DISCIPLINA NOS ENCONTRAMOS?



## 1.1. ANÁLISIS ORIENTADO A OBJETOS



### Objetivo

Consiste en refinar los requisitos funcionales descritos en el Modelo de Casos de Uso (MCU), con el objetivo de comprender de manera más precisa el comportamiento del sistema.

Este proceso permite obtener una representación estructurada y mantenible, que facilita la transición hacia el diseño detallado y la implementación del sistema completo.



## 1.2. ACTIVIDADES DEL ANÁLISIS OO

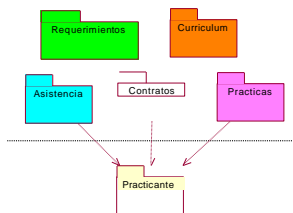


Durante el análisis orientado a objetos, se desarrollan actividades destinadas a refinar los requisitos del sistema y estructurar su funcionamiento interno desde una perspectiva conceptual:

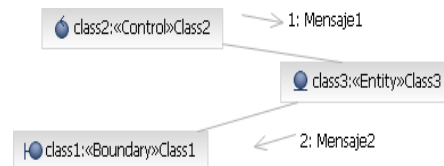
1. Identificación progresiva de nuevos paquetes, clases y requisitos comunes, a medida que se profundiza en la comprensión del sistema.
2. Refinamiento y mantenimiento continuo de los paquetes de análisis, con el objetivo de mejorar la cohesión interna y reducir el acoplamiento entre módulos..

*Se realizan las siguientes actividades*

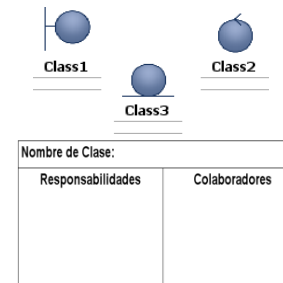
### Análisis de la Arquitectura



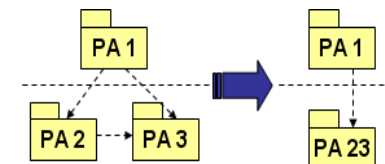
### Análisis de los Casos de Uso



### Análisis de Clases

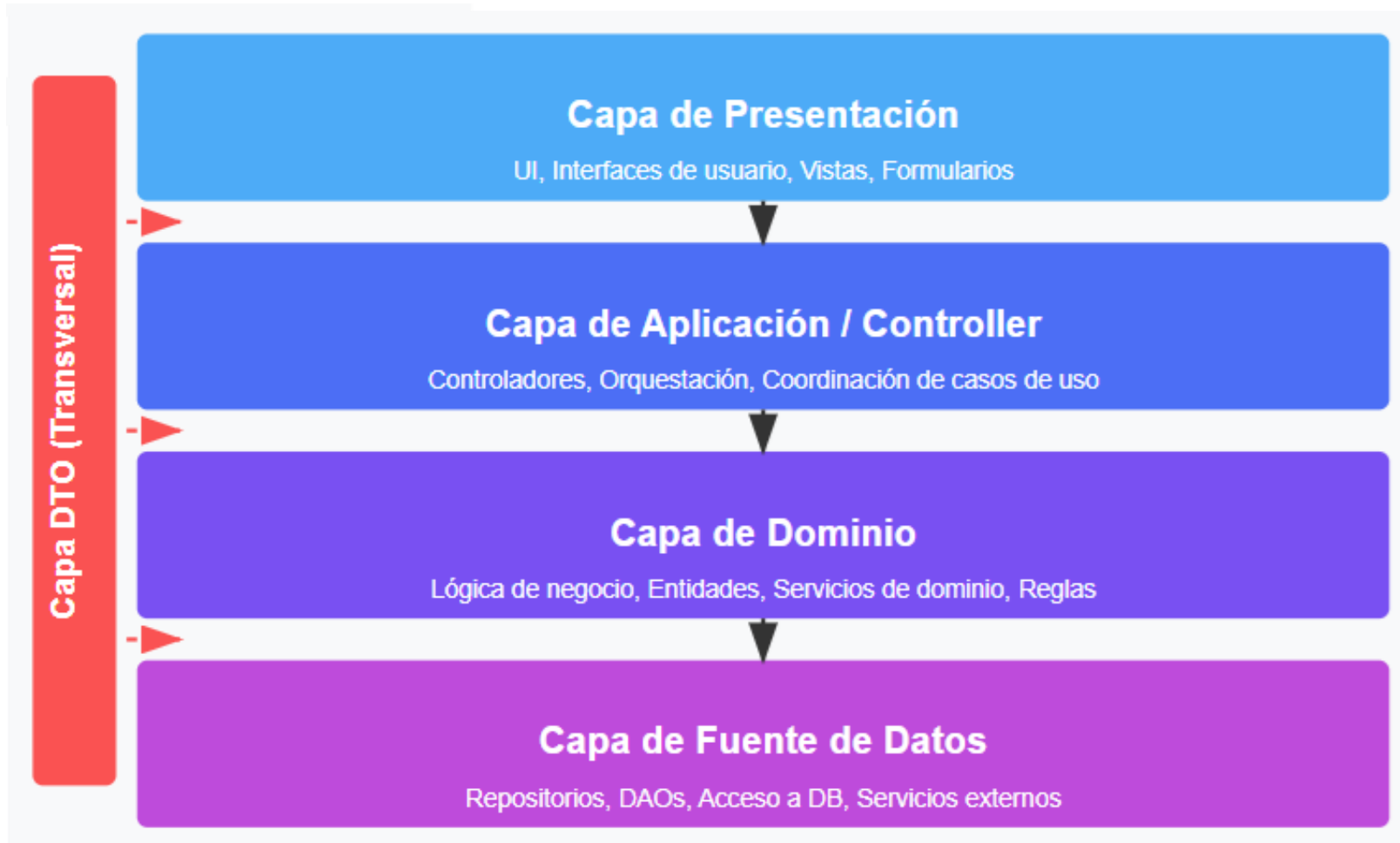


### Análisis de paquetes



## 2.4. LA CAPA DE APLICACIÓN

Arquitectura en Capas con DTOs



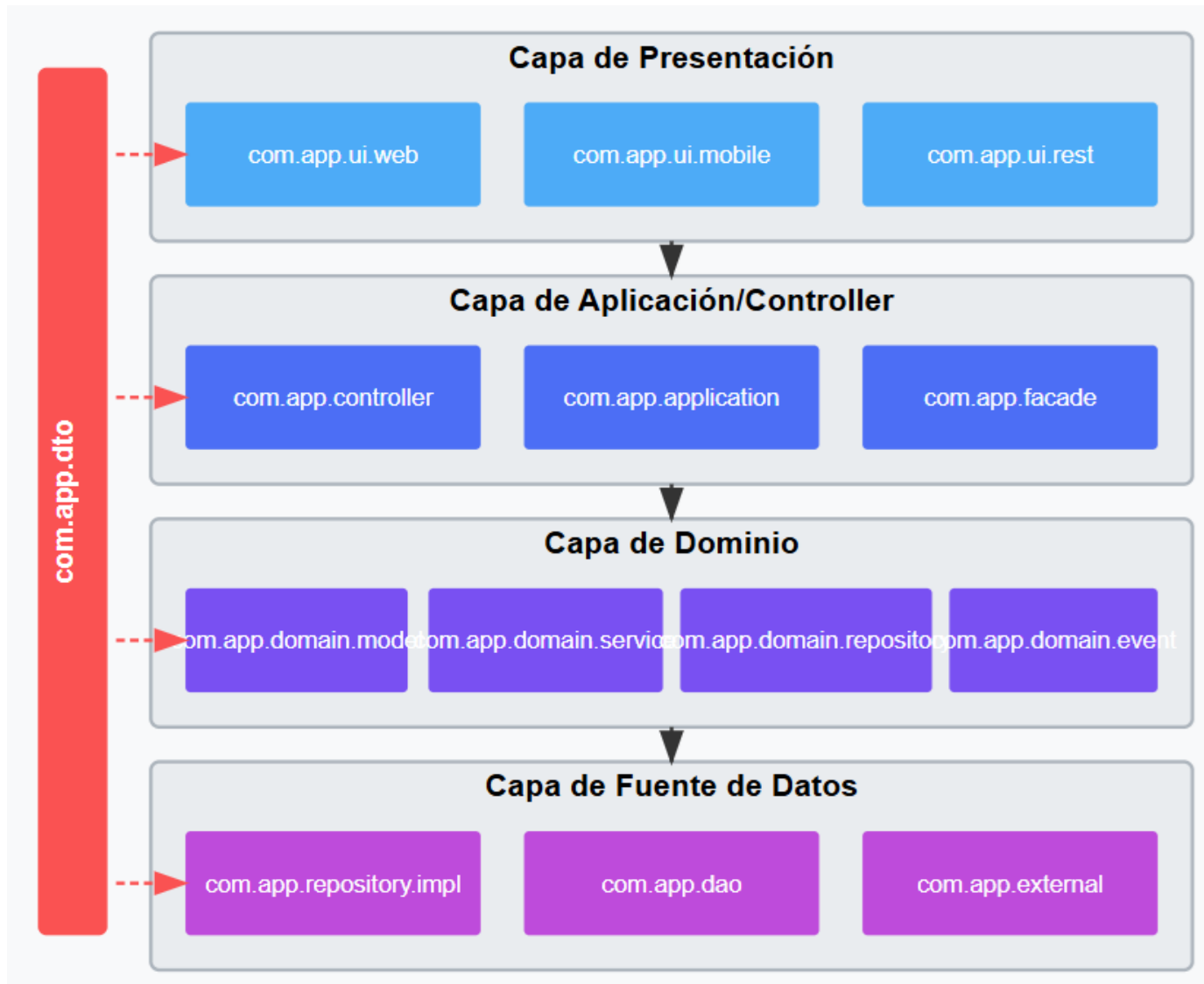
### Leyenda

■ Flujo principal

■ Transferencia de DTOs

## 2.4. LA CAPA DE APLICACIÓN

### Arquitectura en Capas a Nivel de Paquetes

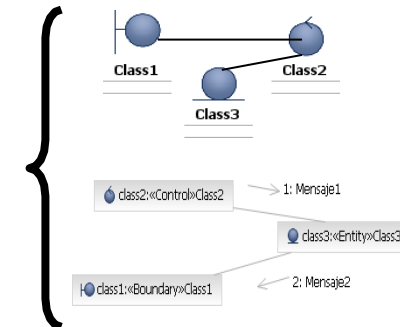
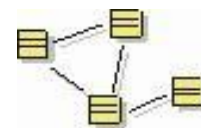
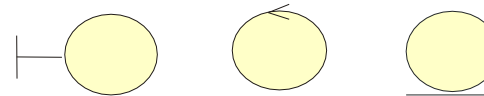
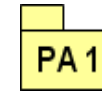
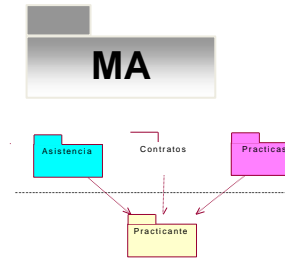


Cada paquete representa una unidad de código cohesiva.

# 1.3. ARTEFACTOS DEL ANÁLISIS OO



- Modelo de análisis
- Arquitectura de análisis
- Paquetes de análisis
- Clases de análisis
- Realizaciones de análisis
- Modelo Conceptual



## 1.4. MODELO DE ANÁLISIS



es

Usado para representar la estructura global del sistema

*Su utilidad radica en que*



Permite una ***apreciación global conceptual del sistema*** porque es un primer intento por definir los conceptos claves que describen el sistema.

## 1.5. Modelo de Casos de Uso vs. Modelo de Análisis



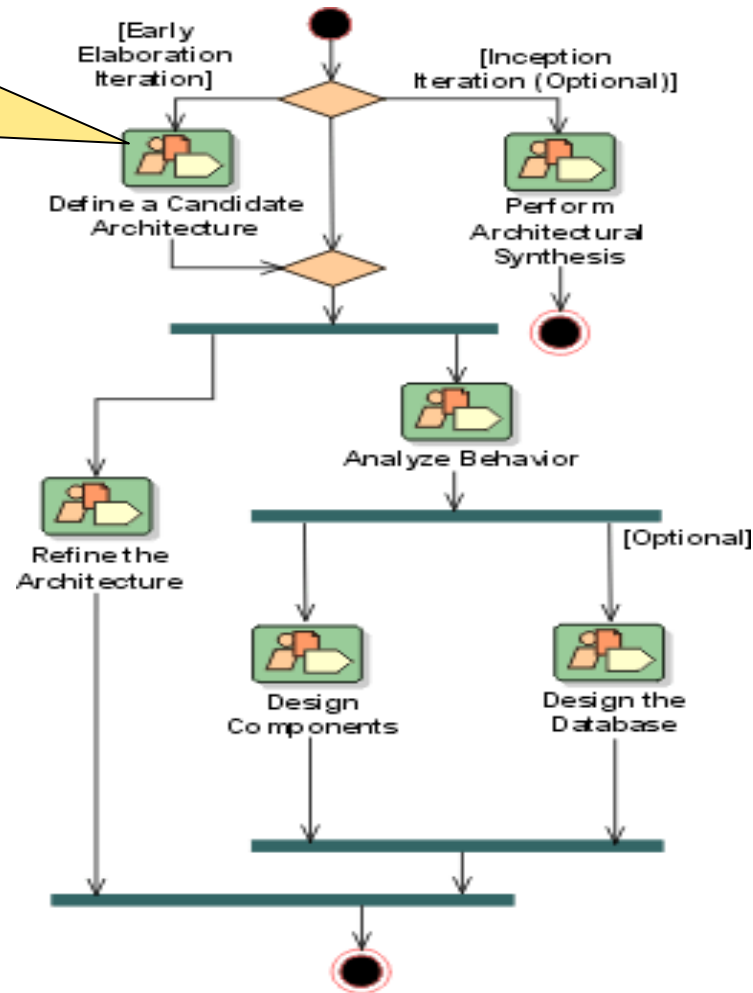
Modelo de Casos de Uso	Modelo de Análisis
Descrito con el lenguaje del cliente. Vista <b>Vista externa del sistema.</b>	Descrita en el lenguaje de los desarrolladores. Vista interna del s <b>Vista interna del sistema.</b>
Estructurado por los c <b>casos de uso.</b>	Estructurado por c <b>clases estereotipadas y paquetes.</b>
Utilizado como contrato entre el cliente y los desarrolladores sobre q <b>qué debería y que NO debería hacer el sistema.</b>	Utilizado por los desarrolladores para comprender c <b>cómo debería darse forma al sistema.</b>
<b>Puede contener redundancias, inconsistencias</b> entre los requisitos.	<b>No debería contener redundancias, inconsistencias</b> entre los requisitos.
Captura la f <b>funcionalidad del sistema.</b>	Esboza como llevar a cabo la f <b>funcionalidad dentro del sistema.</b>
Define los c <b>casos de uso.</b>	Define las r <b>realizaciones de casos de uso.</b>



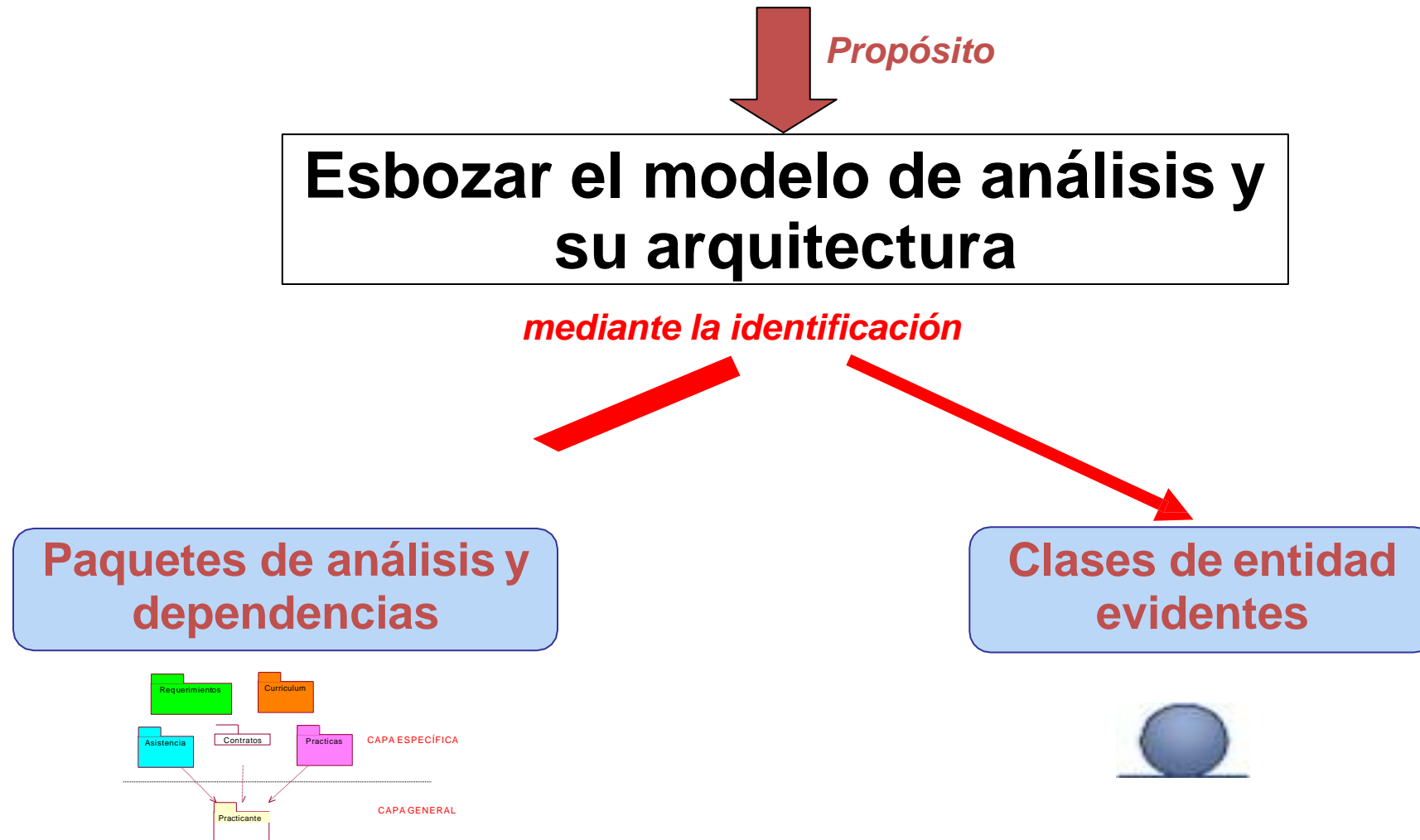
## II. ¿QUÉ ACTIVIDAD DESARROLLAREMOS ?



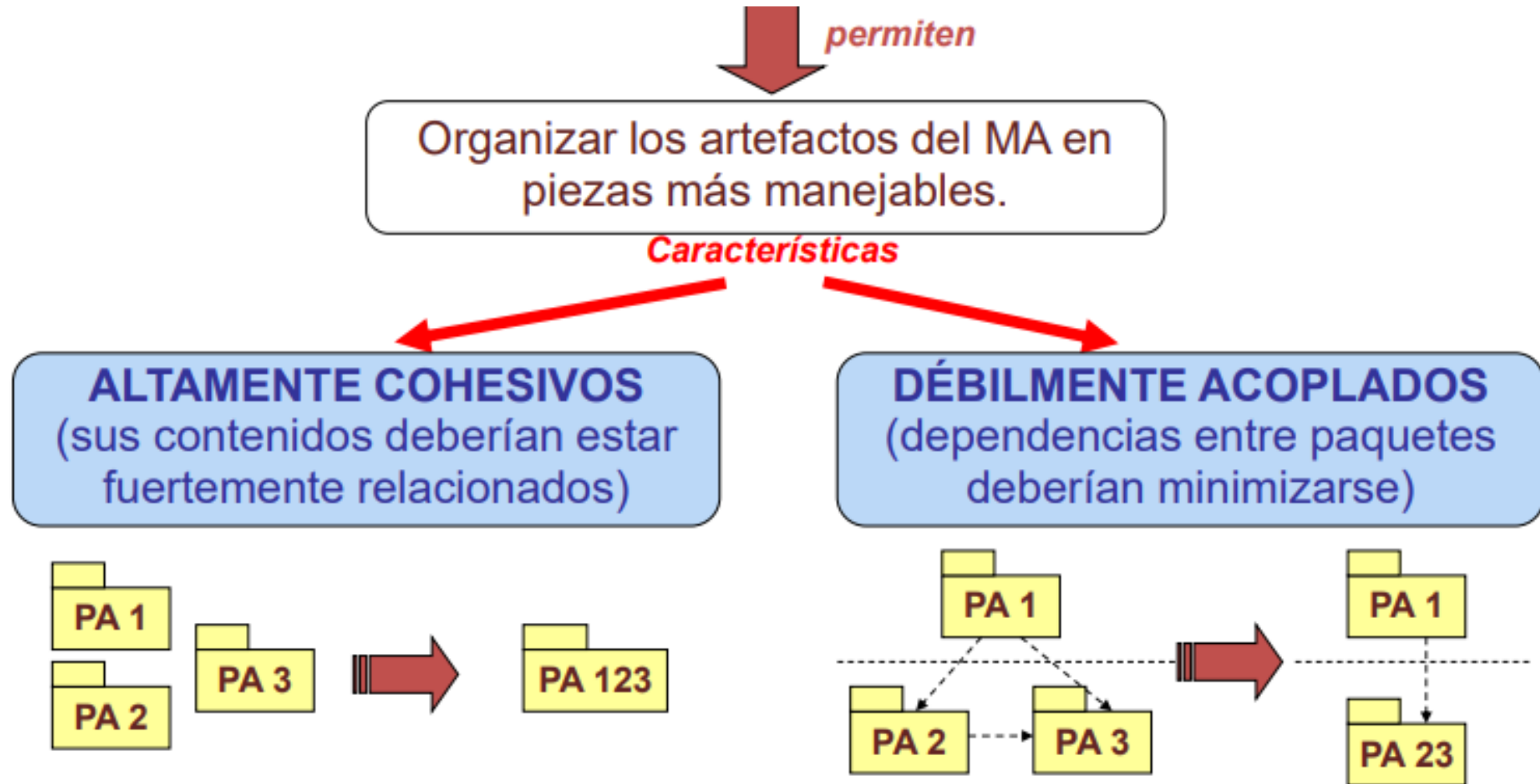
*Esfuerzo en **crear la arquitectura inicial** del sistema (punto inicial para todo el trabajo de análisis).*



## 2.1. ANÁLISIS DE LA ARQUITECTURA



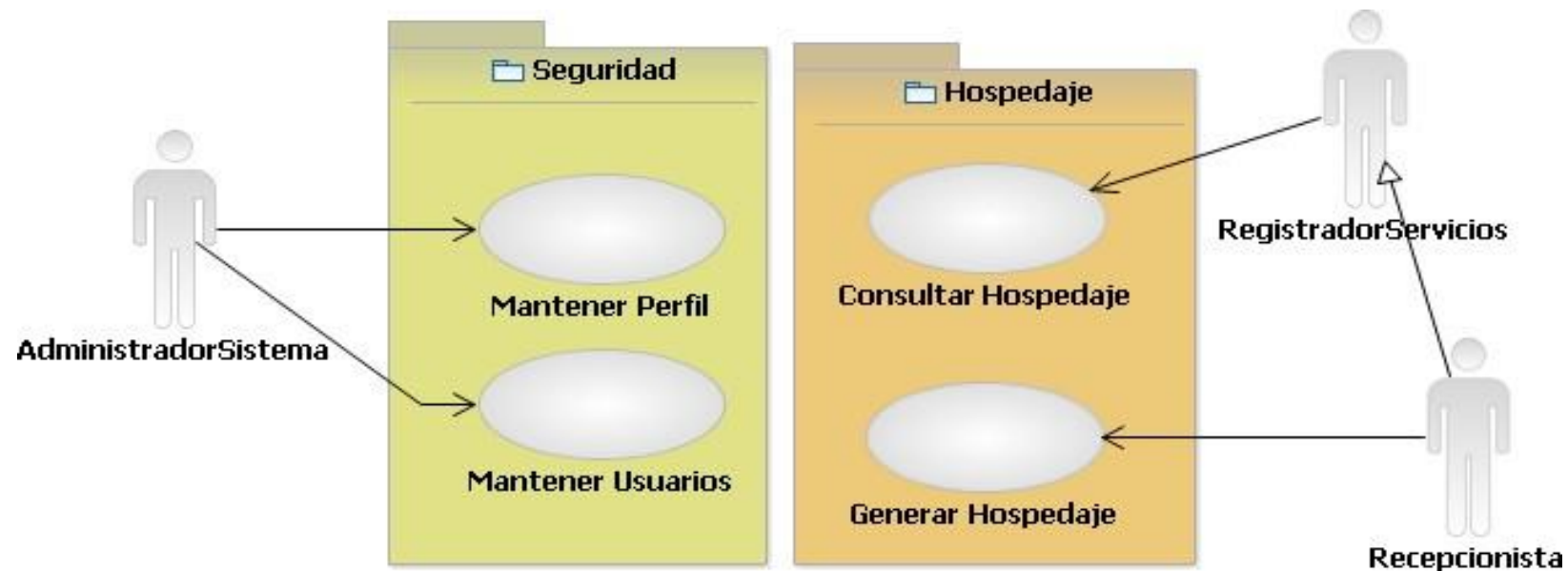
## 2.2. PAQUETES DE ANÁLISIS



## 2.3. IDENTIFICACIÓN DE P. A.



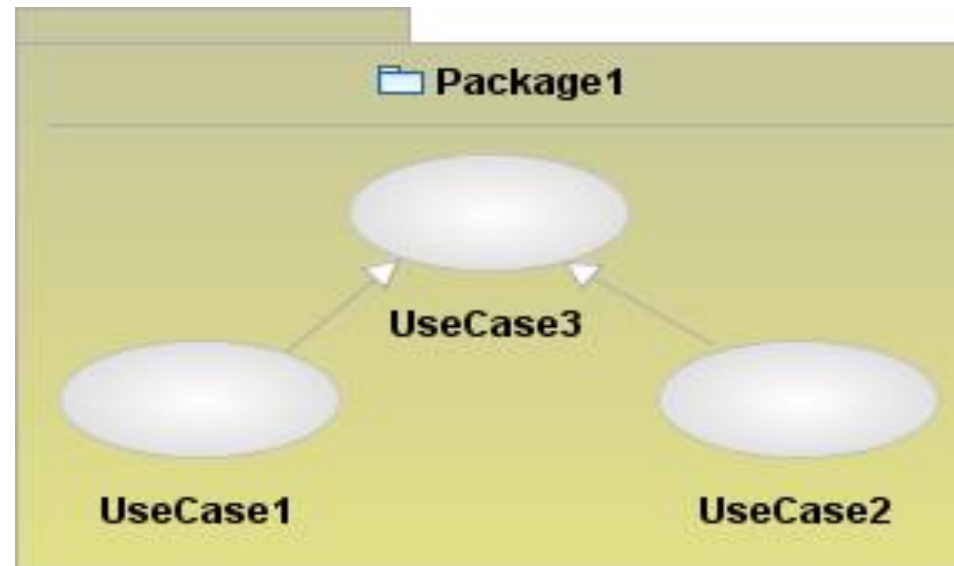
**CONTENIDOS ESTRECHAMENTE RELACIONADOS  
EN UN MISMO PAQUETE**  
(para dar soporte a un determinado actor o a un proceso de negocio)



## 2.3. IDENTIFICACIÓN DE P. A. (CONT.)



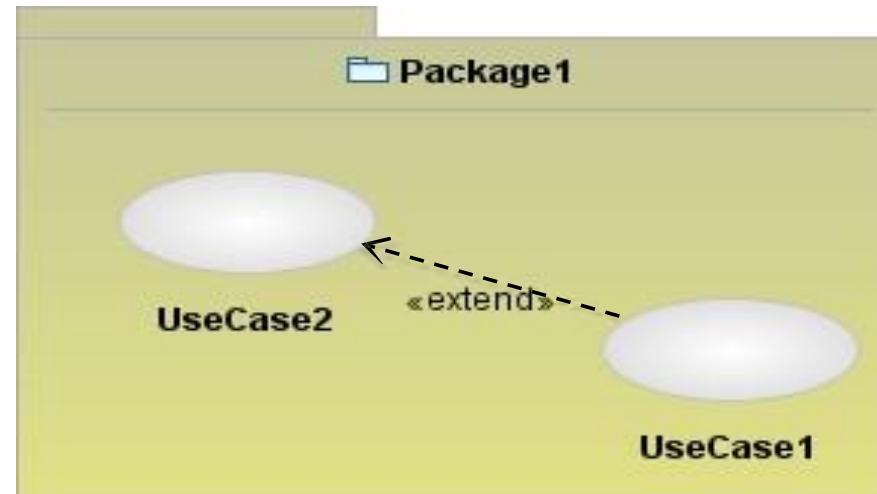
### CASOS DE USO RELACIONADOS CON GENERALIZACIÓN



## 2.3. IDENTIFICACIÓN DE P. A. (CONT.)



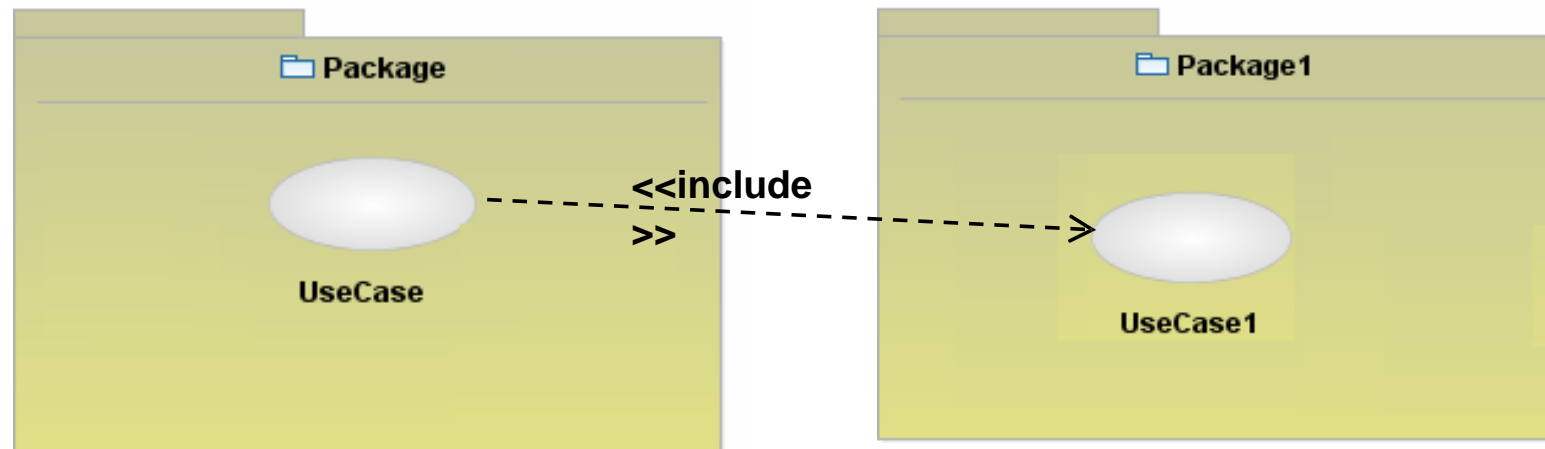
**CASOS DE USO RELACIONADOS CON <<EXTEND>>  
EN UN MISMO PAQUETE**  
(CU extendidos que sólo se extienden a partir de un CU base y  
que no están estrechamente relacionados con CU de otros  
paquetes)



## 2.3. IDENTIFICACIÓN DE P. A. (CONT.)



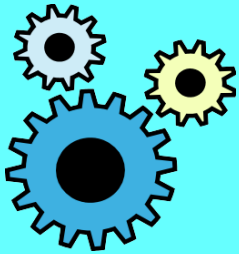
**CASOS DE USO RELACIONADOS CON <<INCLUDE>>  
EN DIFERENTES PAQUETES  
(CU incluidos con contenidos diferentes al CU base)**



## 2.4. LA CAPA DE APLICACIÓN







## Ejercicio

Desarrolle el Diagrama General de CU en el RSA, luego identifique los paquetes de análisis y ubíquelos en la capa correspondiente de la arquitectura de análisis.



La secretaria de Gerencia emite las facturas del equipo de Ventas, asignando a cada factura el vendedor y los productos. Además es la encargada de la administración de los registros de los vendedores y de los Productos. El Administrador posee una opción para anular las facturas y otra opción para dar como pagadas las facturas.

El encargado de almacén tiene una opción para registrar una Solicitud de compra de Productos asignando los productos necesarios a la solicitud, el administrador es el encargado de aprobar las solicitudes de compra de productos.

# PARA RECORDAR



- ❑ Con el AOO se logra comprender de manera más precisa el sistema, refinando la descripción de los requisitos del MCU.
- ❑ El modelo de análisis es un primer intento por definir los conceptos claves que describen el sistema.
- ❑ La utilidad del modelo de análisis radica en que permite una apreciación global conceptual del sistema.



- ✦ El **análisis de la arquitectura** muestra la estructura modular del sistema.
- ✦ La identificación inicial de los PA se hace de manera natural sobre la base de los requisitos funcionales y el dominio del problema.
- ✦ Los PA se organizan en la capa de aplicación, la cual tiene dos capas internas: Específica y General.



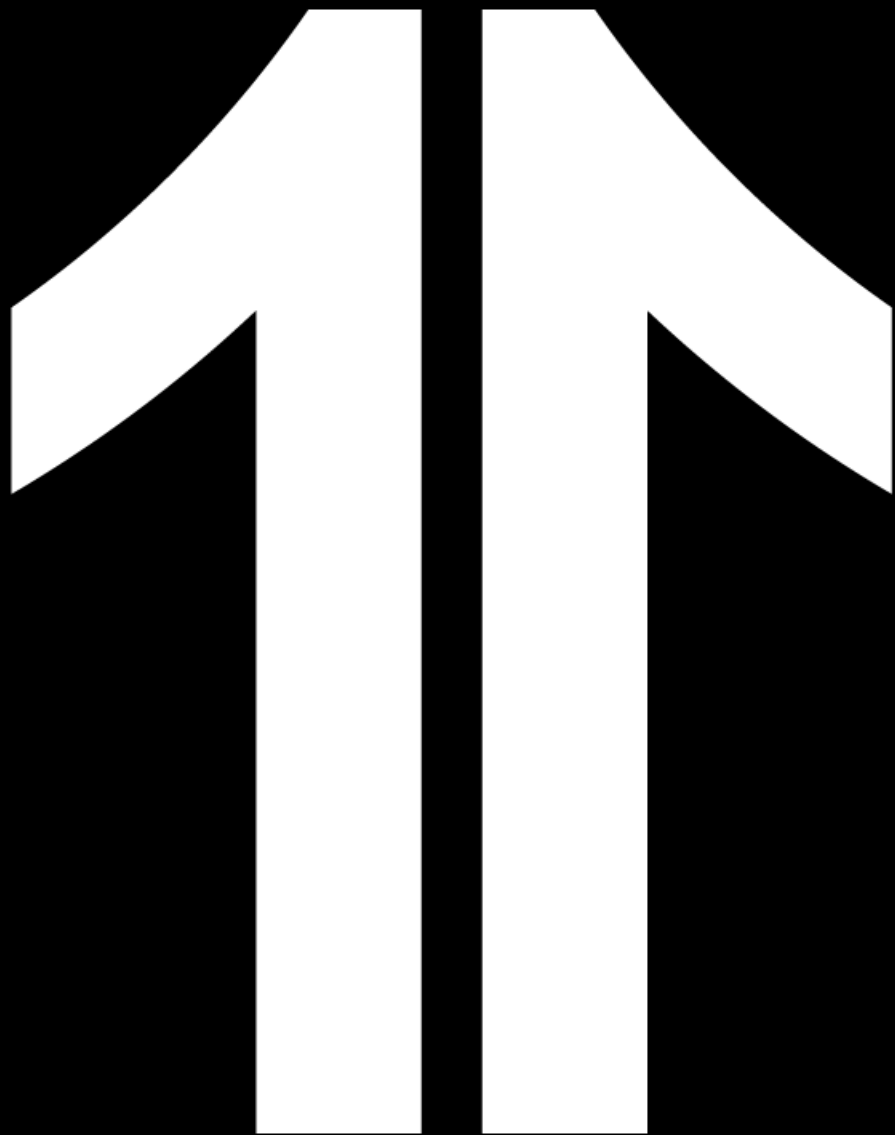
## Cuestionario en la plataforma

Nivel	Rango
Nivel 4	17 – 20
Nivel 3	13 – 16
Nivel 2	9 – 12
Nivel 1	0 - 8

# Comprobación del Logro



Criterio	Nivel 4 (Excelente)	Nivel 3 (Bueno)	Nivel 2 (Regular)	Nivel 1 (Deficiente)
Comprensión conceptual	Demuestra comprensión completa de los conceptos clave.	Comprende la mayoría de los conceptos clave.	Muestra comprensión parcial, con algunas confusiones.	No demuestra comprensión clara de los conceptos.
Aplicación de conocimientos	Aplica correctamente los conceptos en ejemplos o escenarios.	Aplica conceptos con algunos errores menores.	Aplica de forma superficial o con errores evidentes.	No logra aplicar los conceptos o lo hace incorrectamente.
Claridad y precisión en respuestas	Las respuestas son claras, bien estructuradas y precisas.	Las respuestas son mayormente claras, con leves imprecisiones.	Respuestas poco claras o con ideas mal organizadas.	Respuestas confusas, incompletas o incoherentes.
Cobertura del contenido esperado	Responde completamente las 5 preguntas, desarrollando bien cada una.	Responde correctamente 4 de las 5 preguntas.	Responde correctamente 2 o 3 preguntas.	Responde solo 1 pregunta correctamente o no responde.



**GRACIAS**

