

DISEÑO DE LOS CASOS DE USO

Introducción

Patrones de diseño para asignar responsabilidades

Patrones GRASP

Elaboración del modelo de interacción de objetos

DISEÑO DE LOS CASOS DE USO

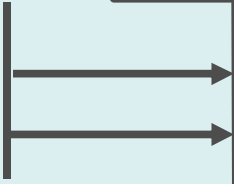
Introducción

Modelo de Análisis

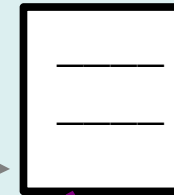
Diagrama conceptual



:Sistema



Contrato de las operaciones



Herramienta:
Diagrama de
interacción de UML

Modelo de interacción de objetos

Ayuda:
Patrones para la
asignación de
responsabilidades a
objetos

DISEÑO DE LOS CASOS DE USO

Patrones de diseño para asignar responsabilidades

Responsabilidad

Contrato u obligación que debe tener un objeto en su comportamiento

Categorías

Hacer



- ✚ Hacer algo en uno mismo
- ✚ Iniciar una acción en otros objetos
- ✚ Controlar y coordinar actividades en otros objetos

Conocer



- ✚ Estar enterado de los datos privados encapsulados
- ✚ Estar enterado de la existencia de objetos conexos
- ✚ Estar enterado de las cosas que se pueden derivar o calcular

Responsabilidad ≠ Método

→ Cumplen las responsabilidades

→ Se implementan usando métodos que operan solos o en colaboración con otros métodos

DISEÑO DE LOS CASOS DE USO

Patrón de diseño

Descripción de un problema con su solución en un determinado contexto

Son una forma de reutilizar el conocimiento y la experiencia de otros diseñadores

Partes esenciales de un patrón

✚ Nombre

Que sea una referencia significativa al patrón

✚ Problema

Una descripción del problema que enuncie cuándo se puede aplicar el patrón

✚ Solución

Una descripción de la solución de diseño, sus relaciones y responsabilidades. Es una plantilla para que una solución se instale en diferentes formas

✚ Consecuencias▶ ayudan a entender si es factible usar el patrón en una situación particular

Los resultados (buenos y malos) y las negociaciones al aplicar el patrón

DISEÑO DE LOS CASOS DE USO

Patrones GRAPS

General Responsibility Assignment Software Patterns

Describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones

Características

- No expresan nuevos principios de la ingeniería del software
 - Codifican conocimiento, expresiones y principios ya **existentes**
 - Son un ejemplo de fuerza de abstracción porque dan nombre a una idea compleja
- Apoyan la incorporación de conceptos a nuestro sistema cognitivo y a la memoria
- Facilitan la comunicación

Patrones GRAPS

Experto en Información
Creador
Bajo Acoplamiento
Alta cohesión
Controlador

Polimorfismo
Fabricación Pura
Indirección
No hables con extraños
...

DISEÑO DE LOS CASOS DE USO

Experto en Información

Problema

¿Cuál es el principio general para asignar responsabilidades a los objetos?

Solución

Asignar responsabilidad a la clase que contiene la información necesaria para llevarla a cabo

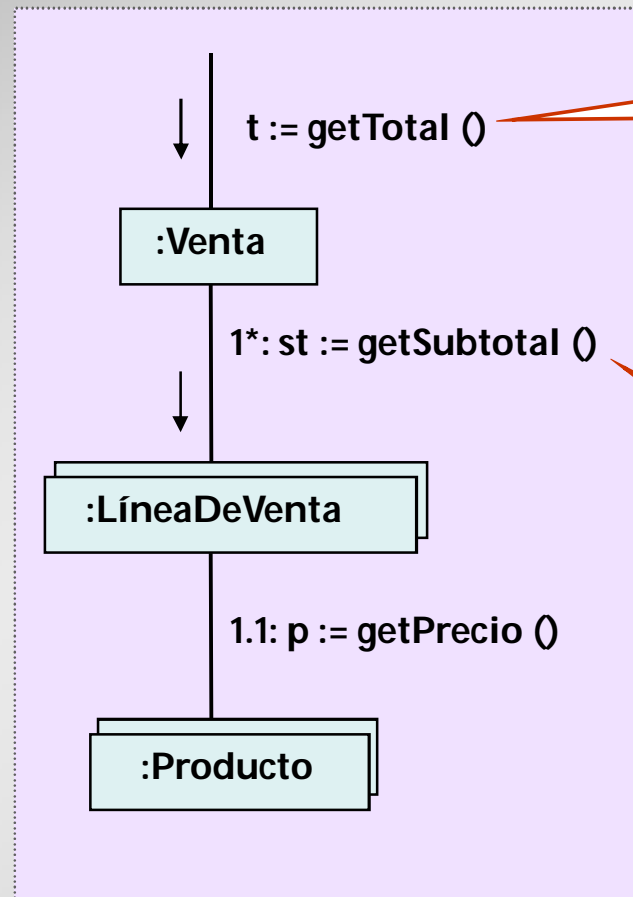
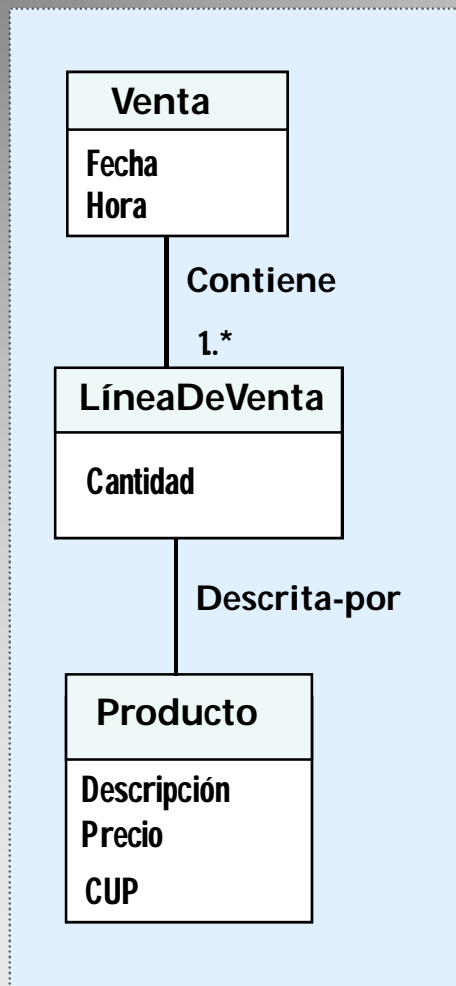
Consecuencias

MALAS: En ocasiones va en contra de los principios de acoplamiento o cohesión

BUENAS: Mantiene el ocultamiento de información y distribuye el comportamiento

DISEÑO DE LOS CASOS DE USO

Ejemplo



¿Quién conoce el total de una Venta?

¿Qué información se necesita para calcular el subtotal de la línea de venta?

DISEÑO DE LOS CASOS DE USO

Creador

Problema

¿Quién debería ser el responsable de la creación de una instancia de alguna clase?

Solución

Asignar a la clase B la responsabilidad de crear instancias de A cuando:

- + B agrega objetos de A
- + B contiene objetos de A
- + B registra objetos de A
- + B utiliza objetos de A
- + B tiene los datos de inicialización de A

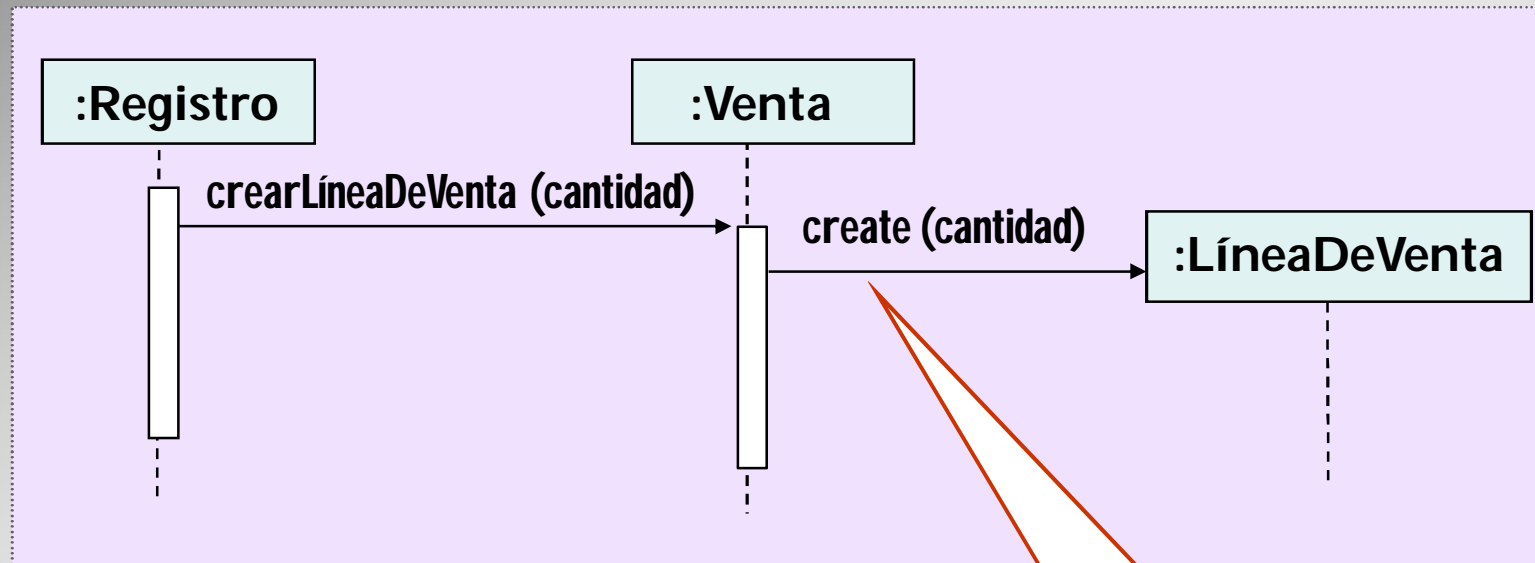
Consecuencias

MALAS: No es conveniente su uso cuando se construye a partir de instancias existentes

BUENAS: Produce bajo acoplamiento

DISEÑO DE LOS CASOS DE USO

Ejemplo



¿Quién debería ser el responsable de crear una instancia de **LíneaDeVenta**?

DISEÑO DE LOS CASOS DE USO

Bajo Acoplamiento

Problema

¿Cómo soportar bajas dependencias, bajo impacto del cambio e incremento de la reutilización?

Solución

Asignar una responsabilidad de manera que el acoplamiento permanezca bajo

Consecuencias

MALAS: Llevado a extremo puede ocasionar diseños pobres; en un conjunto de clases debe haber un nivel de acoplamiento moderado y adecuado

BUENAS: No afectan los cambios en otros componentes

Fáciles de entender de manera aislada

Conveniente para reutilizar

DISEÑO DE LOS CASOS DE USO

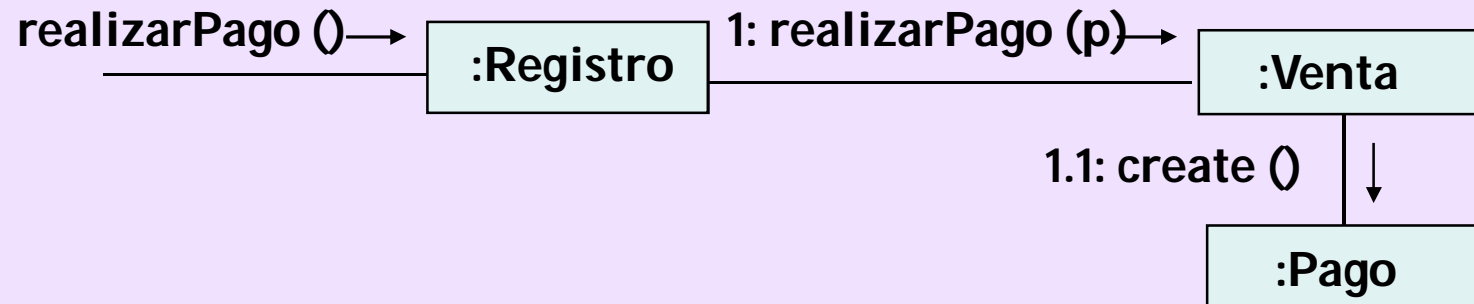
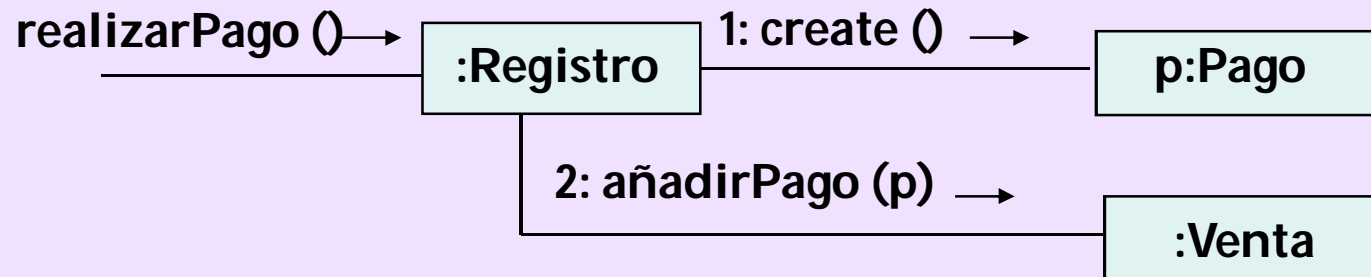
Ejemplo

Pago

Registro

Venta

¿Qué clase debería ser responsable de crear una instancia Pago y asociarla con la Venta?



¿Qué diseño soporta Bajo Acoplamiento?

DISEÑO DE LOS CASOS DE USO

Formas comunes de acoplamiento

- El **Tipo X** tiene un atributo que referencia a una instancia de **Tipo Y**, o al propio **Tipo Y**
- Un objeto de **Tipo X** invoca los servicios de un objeto de **Tipo Y**
- El **Tipo X** tiene un método que referencia a una instancia de **Tipo Y**, o al propio **Tipo Y**, de algún modo
- El **Tipo X** es una subclase, directa o indirecta, del **Tipo Y**
- El **Tipo Y** es una interfaz y el **Tipo X** implementa esa interfaz

DISEÑO DE LOS CASOS DE USO

Alta Cohesión

Problema

¿Cómo mantener la complejidad manejable?

Solución

Asignar una responsabilidad de manera que la cohesión permanezca alta

Consecuencias

MALAS: Ninguna, renunciar a la alta cohesión sólo cuando esté muy justificado

BUENAS: Claridad y facilidad de entendimiento del diseño

Simplificación del mantenimiento y de las mejoras

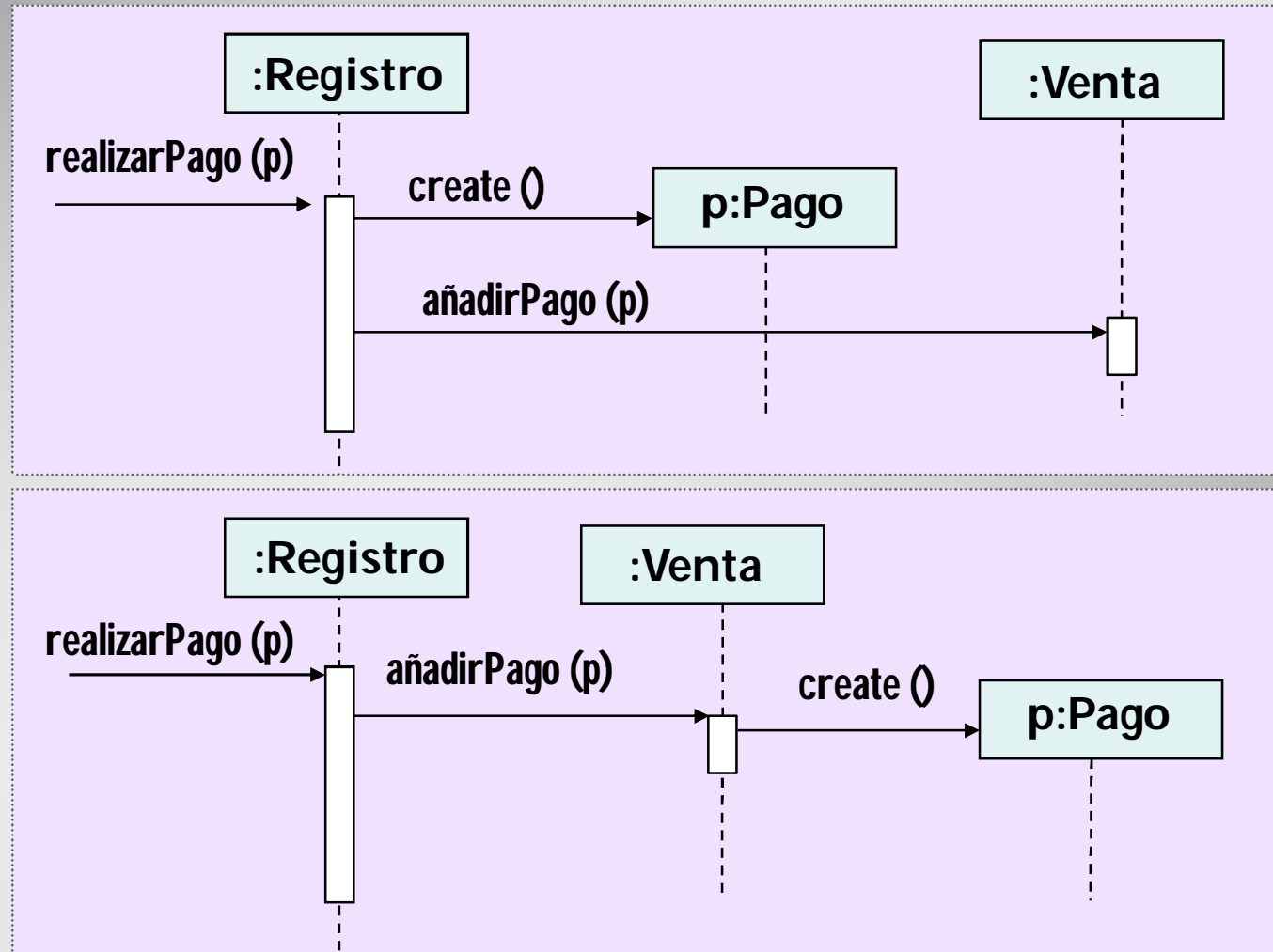
A menudo soporta bajo acoplamiento

Incremento de la reutilización

DISEÑO DE LOS CASOS DE USO

Ejemplo

¿Qué clase debería ser responsable de crear una instancia de Pago y asociarla con la Venta?



¿Qué diseño soporta Alta Cohesión?

DISEÑO DE LOS CASOS DE USO

Grados de cohesión funcional

■ Muy baja cohesión

Una única clase es responsable de muchas cosas en áreas funcionales diferentes

■ Baja cohesión

Una única clase tiene la responsabilidad de una tarea compleja en un área funcional

■ Alta cohesión

Una única clase tiene una responsabilidad moderada en un área funcional y colabora con otras clases para llevar a cabo las tareas

■ Moderada cohesión

Una única clase tiene responsabilidades ligeras y únicas en unas pocas áreas diferentes que están lógicamente relacionadas con el concepto de la clase, pero no entre ellas

DISEÑO DE LOS CASOS DE USO

Controlador

Problema

¿Quién debe ser responsable de gestionar un **evento de entrada al sistema**?

→ Generado por un actor externo
→ Se asocia con operaciones del sistema

Solución

Asignar la responsabilidad de recibir o manejar un mensaje de evento del sistema a una clase que represente

- ✚ El sistema global, dispositivo o subsistema (controlador de fachada)
- ✚ El escenario de caso de uso en el que tiene lugar el evento del sistema (controlador de caso de uso)

Consecuencias

MALAS: Controladores saturados

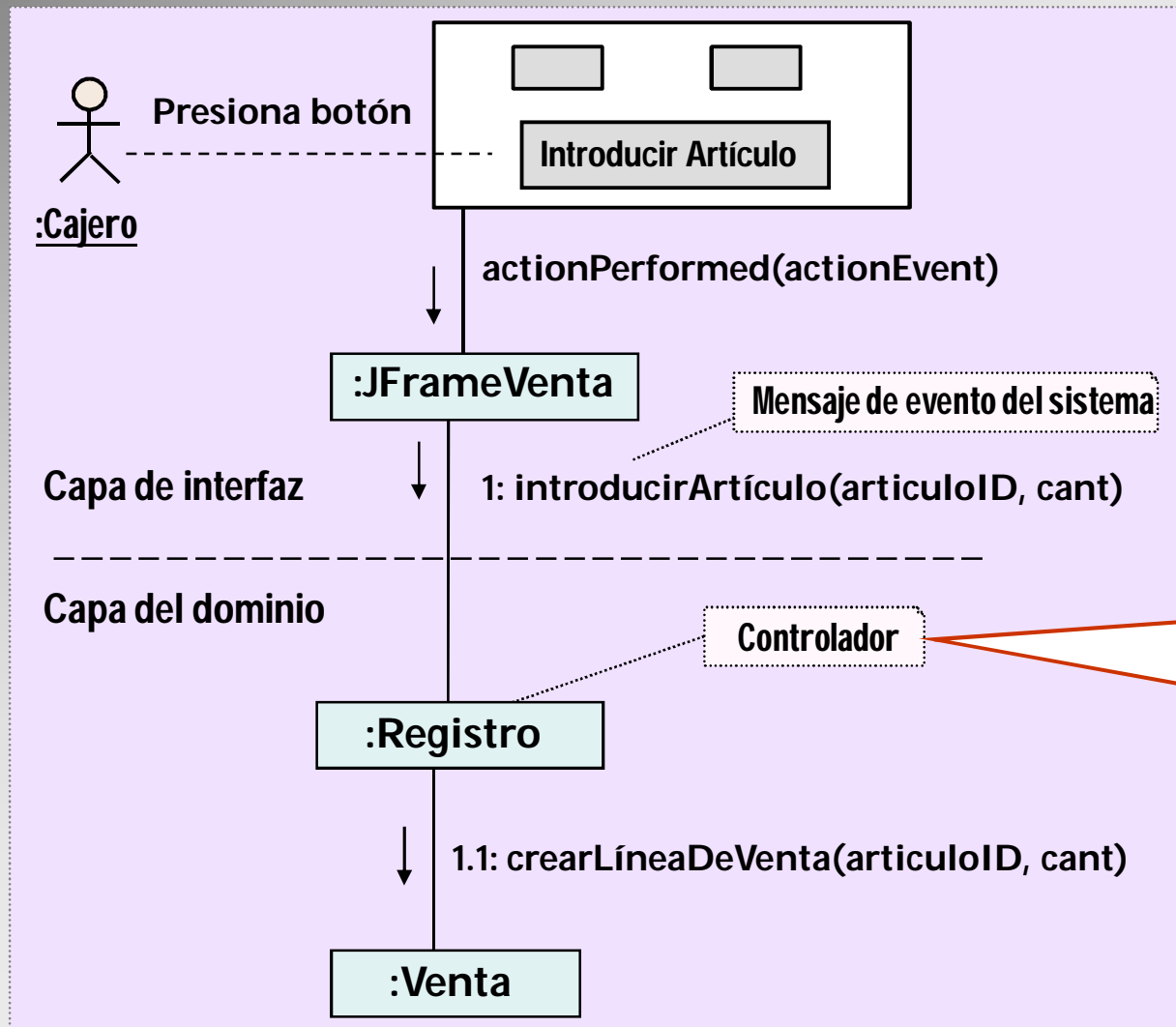
BUENAS: Se asegura que la lógica de la aplicación no se maneje en la interfaz

Aumento de la reutilización y bajo nivel de acoplamiento

Posibilidad de razonar sobre el estado de los casos de uso

DISEÑO DE LOS CASOS DE USO

Ejemplo



No realiza trabajo, sino que delega
Es una especie de fachada sobre el dominio para la interfaz

DISEÑO DE LOS CASOS DE USO

Elaboración del modelo de interacción de objetos

Directrices generales

- Las bases principales para obtener los diagramas de interacción son los **contratos** y el **modelo conceptual**
- El modelo conceptual sirve de guía para saber qué objetos pueden interaccionar en una operación
- Todo lo especificado en el contrato, especialmente las poscondiciones, excepciones y salidas, tienen que satisfacerse en el correspondiente diagrama de comunicación
- Para la elaboración de cada diagrama de comunicación se aplican los **patrones de diseño**

DISEÑO DE LOS CASOS DE USO

Pasos a seguir

A. Elaborar los diagramas de interacción

Para cada operación especificada en los diagramas de secuencia

1. Tener presente el diagrama de conceptos y el contrato de la operación
2. Representar las relaciones del controlador con los objetos que intervienen en la interacción
3. Asignar responsabilidades a objetos
4. Establecer tipos de enlaces entre objetos

B. Inicialización del sistema

C. Establecer relaciones entre el modelo y la Interfaz de Usuario

DISEÑO DE LOS CASOS DE USO

A.1 Considerar el diagrama de conceptos y el contrato de la operación

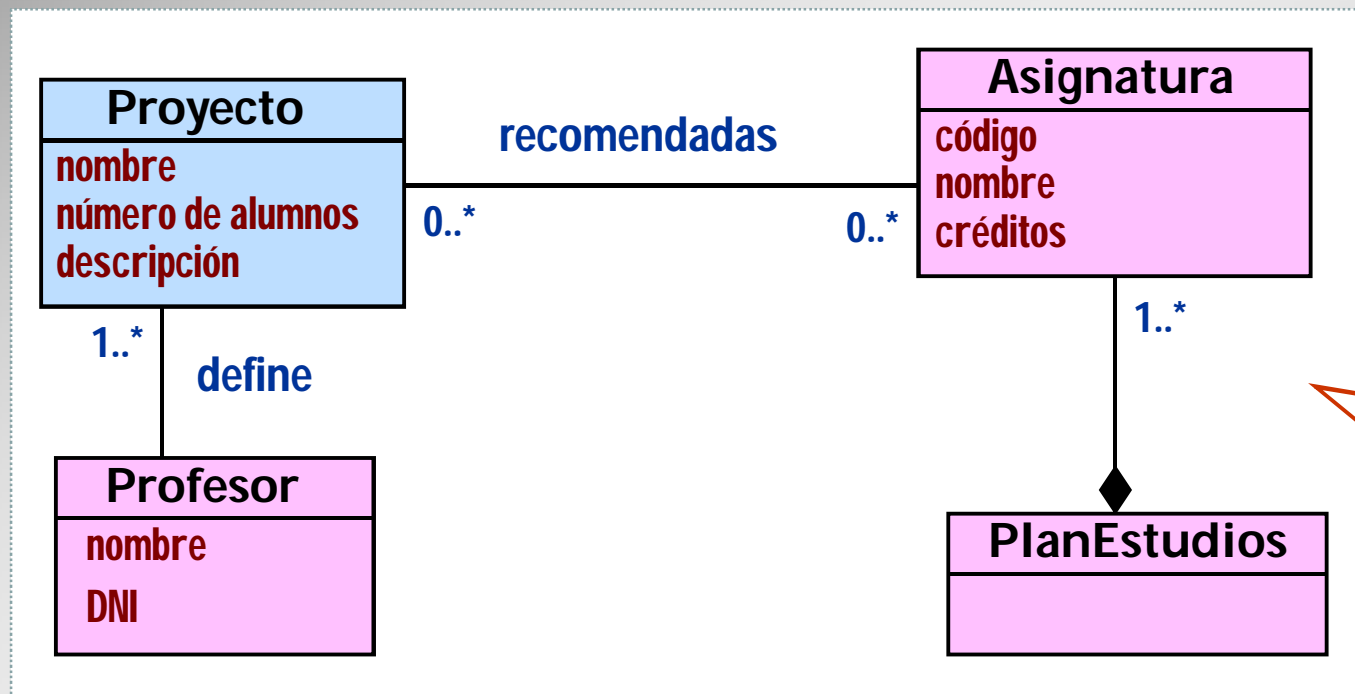
Básico: El contrato de la operación

Nombre	definirProyecto (idProfesor, titulo, numAlum, descrip, listIdAsig)
Responsabilidad	Añade un nuevo proyecto a la lista de proyectos definidos por el profesor y establece sus asignaturas recomendadas del plan de estudios
Ref. cruzadas	<Casos de uso en los que está la operación>
Tipo	SAP
Excepciones	<ul style="list-style-type: none">- Si no existe el profesor identificado por idProfesor- Si $!(0 < \text{numAlum} < 4)$- Si no existe alguna de las asignaturas identificada por el correspondientes elemento de listIdAsig
Poscondiciones	<ul style="list-style-type: none">- Fue creado un objeto Proyecto, pro, debidamente inicializado- Fue creado un enlace entre pro y el objeto Profesor, identificado por idProfesor Para cada elemento de listIdAsig <ul style="list-style-type: none">- Fue creado un enlace entre pro y el objeto Asignatura identificado por el correspondiente elemento de listIdAsig

DISEÑO DE LOS CASOS DE USO

A.1 Considerar el diagrama de conceptos y el contrato de la operación

Complementario: Parte del Diagrama de Conceptos en el que aparecen los conceptos relacionados con los objetos que intervienen en la interacción



Objetos que intervienen en la operación.
En **azul** lo que se modifica y en **rosa** lo que se consulta

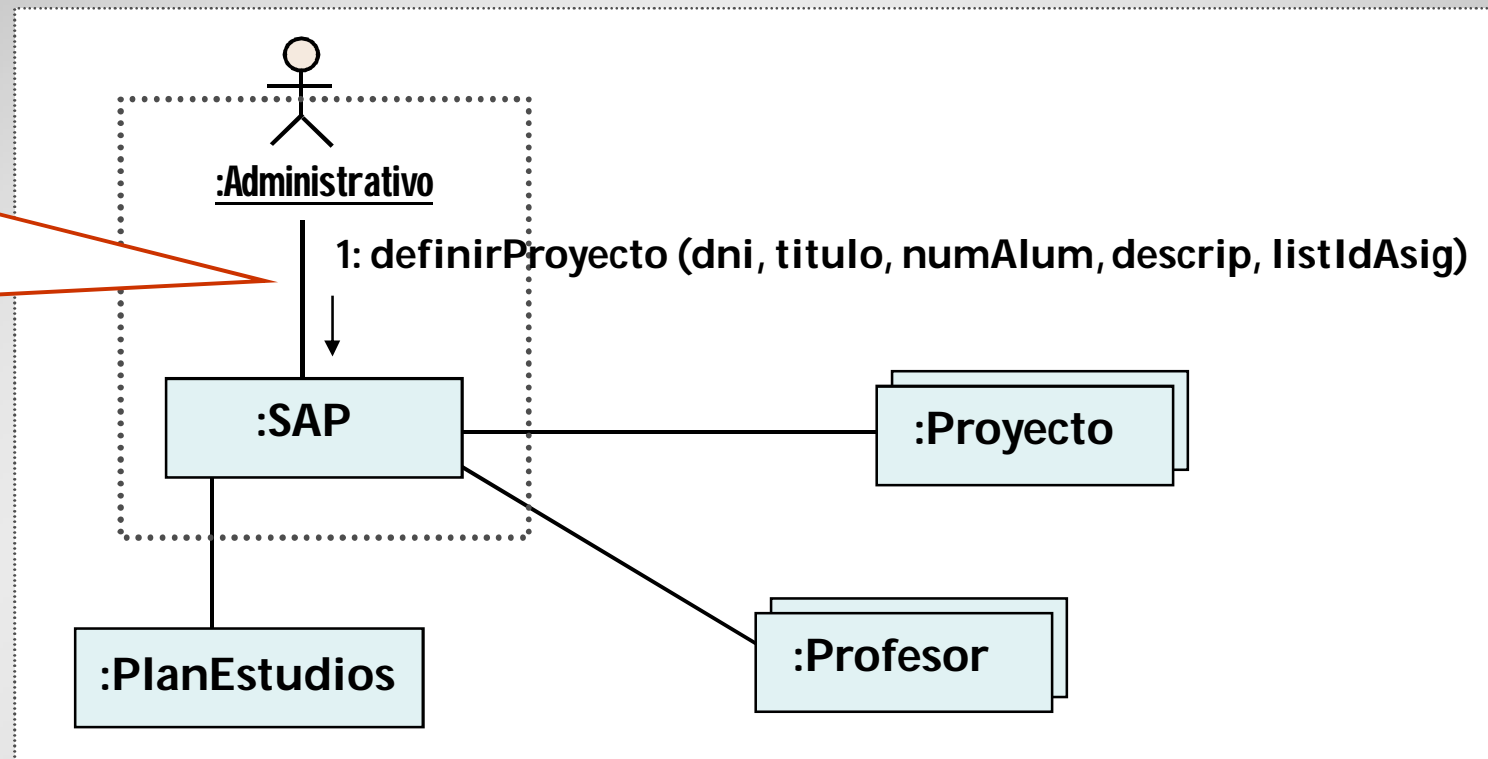
DISEÑO DE LOS CASOS DE USO

A.2 Representar las relaciones del controlador con los objetos que intervienen en la interacción

¿Qué objetos necesita conocer directamente el controlador?

Necesita conocer a todos los profesores, a todos los proyectos y al plan de estudios

Este primer nivel procede del Diagrama de Secuencia del Sistema



DISEÑO DE LOS CASOS DE USO

A.3 Asignar responsabilidades a objetos

Según el nivel en el que se encuentre la elaboración del diagrama, para cada objeto se debe formular la siguiente pregunta:

De todo lo que se dice en el contrato, ¿de qué es responsable?

La respuesta es aplicar los patrones de diseño, fundamentalmente el experto en información y el creador

Se comienza por el controlador: en el ejemplo SAP es responsable de

- + Cumplir con las excepciones (experto en información)

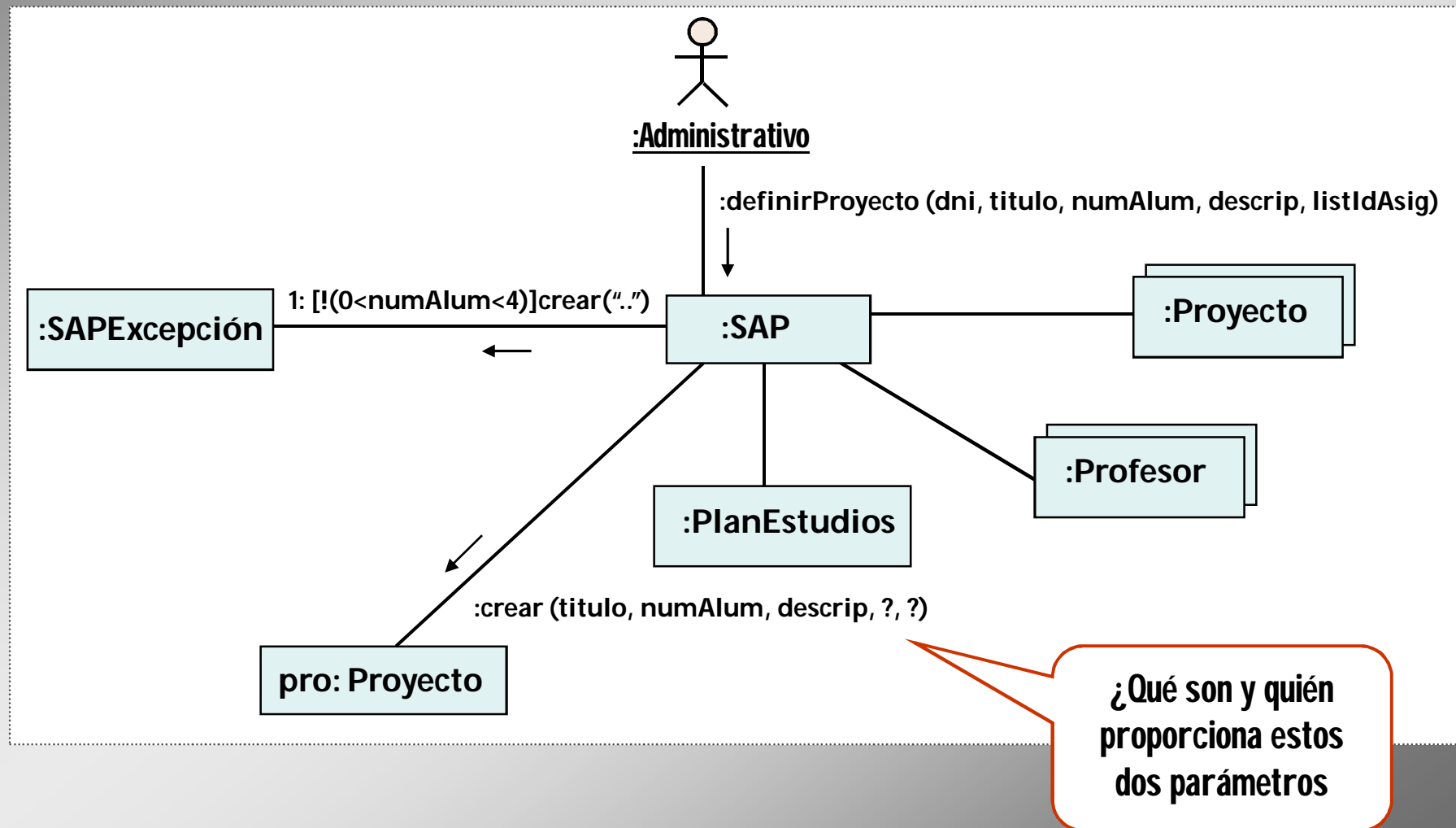
- + Cumplir con la primera poscondición (el creador)

Simplificación:

Las excepciones en las búsquedas de objetos no se van representar en los Diagramas de Comunicación, se dejarán internas a la operación de búsqueda

DISEÑO DE LOS CASOS DE USO

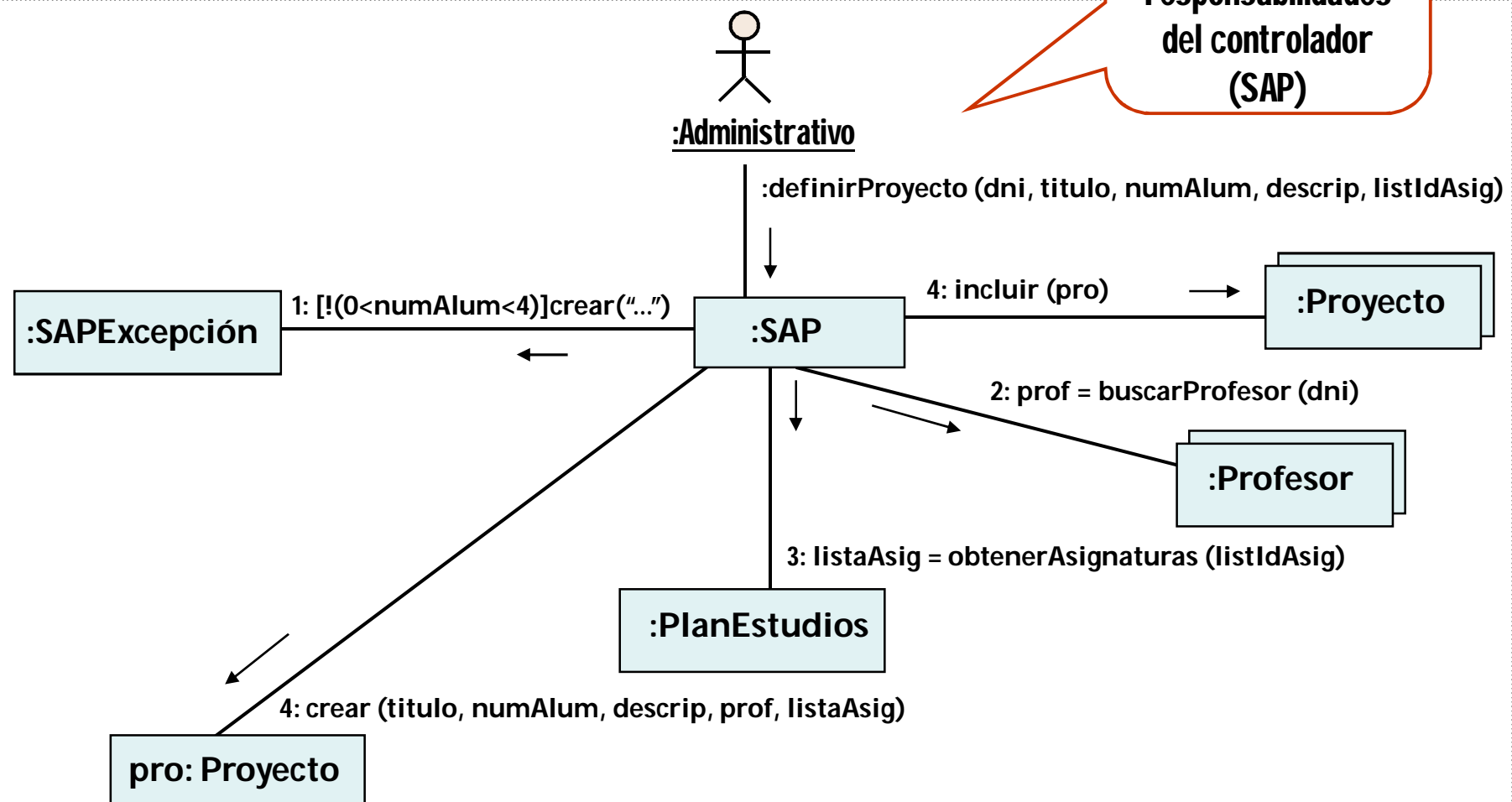
A.3 Asignar responsabilidades a objetos



DISEÑO DE LOS CASOS DE USO

A.3 Asignar responsabilidades a objetos

Aquí están incluidas todas las responsabilidades del controlador (SAP)



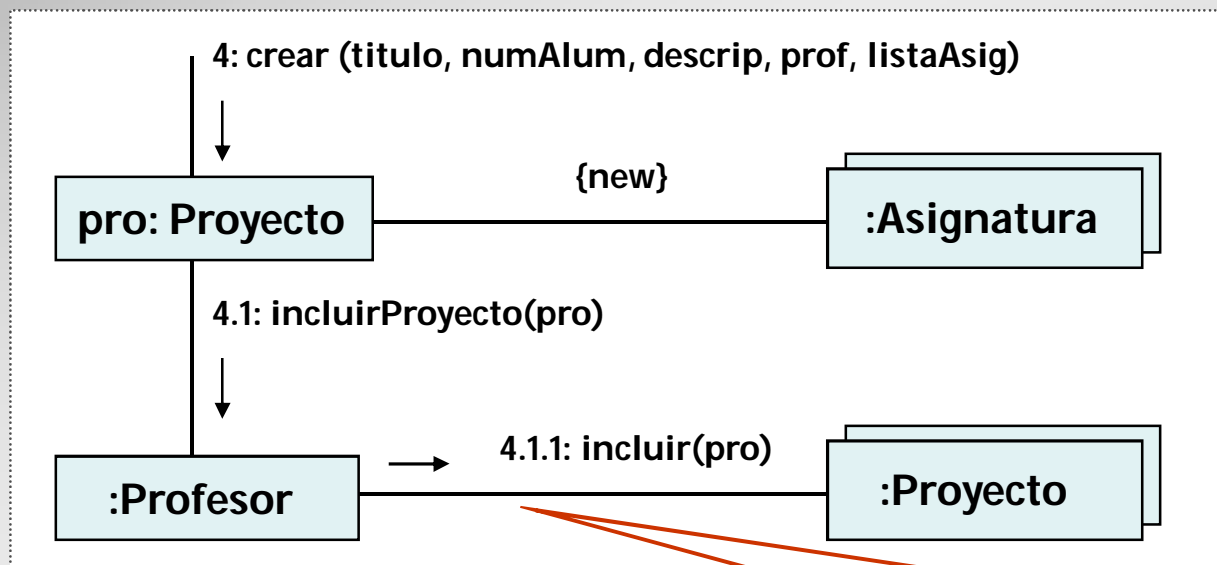
DISEÑO DE LOS CASOS DE USO

A.3 Asignar responsabilidades a objetos

Seguir asignando responsabilidades a los objetos que van apareciendo

En el ejemplo, el nuevo proyecto creado es responsable de:

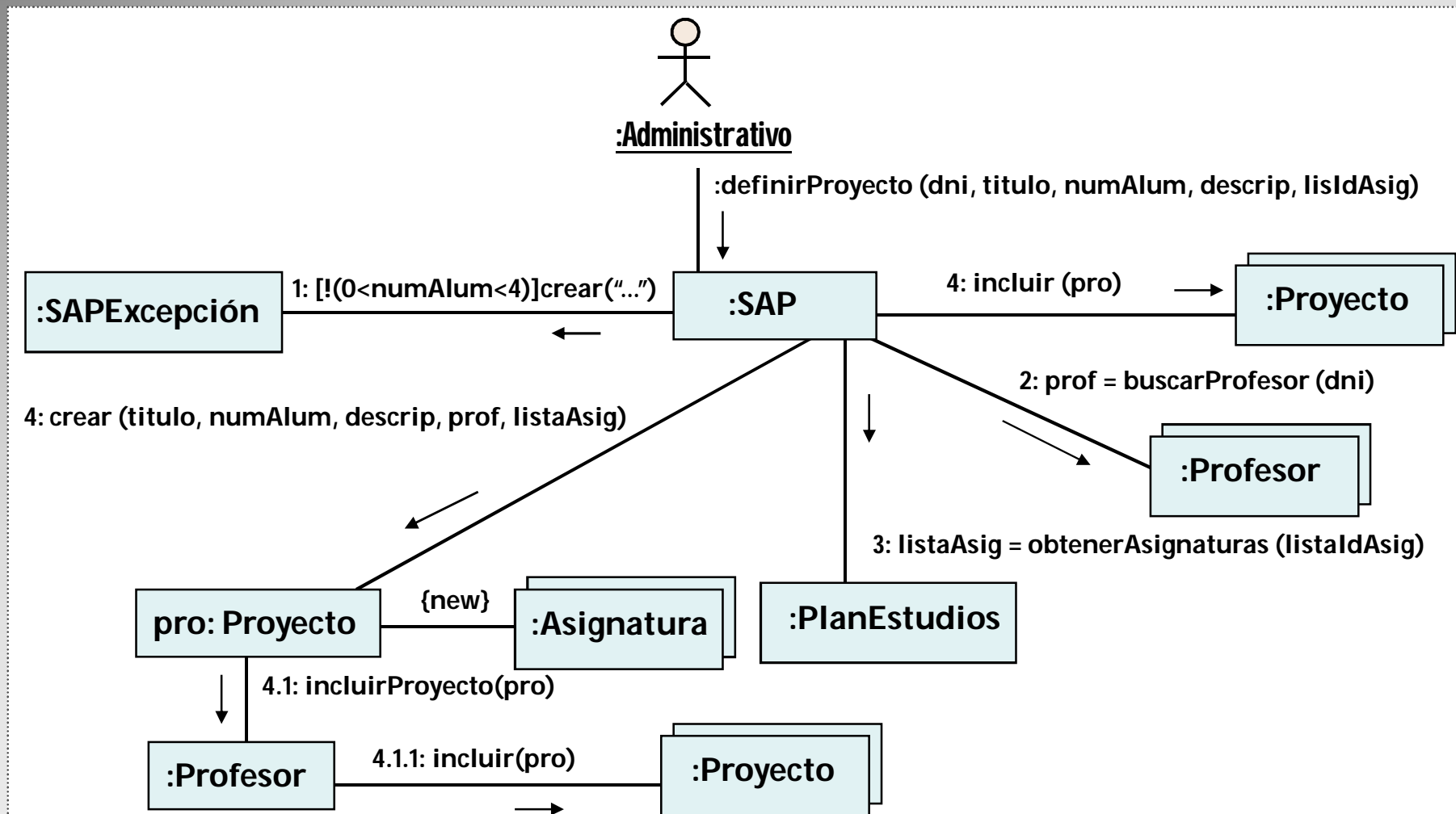
- + Cumplir con la dos últimas poscondiciones, es decir, construir los enlaces correspondiente (experto en información)



Doble enlace, el profesor conoce sus proyectos y el proyecto conoce a su profesor

DISEÑO DE LOS CASOS DE USO

A.3 Asignar responsabilidades a objetos



DISEÑO DE LOS CASOS DE USO

A.4 Establecer tipos de enlaces entre objetos: Estereotipo de visibilidad

Visibilidad es la capacidad de un objeto de “ver” o tener una referencia a otro objeto

■ Visibilidad de atributo

Desde A a B cuando B es un atributo de A

Relativamente permanente porque persiste mientras existan A y B

<<A>>

■ Visibilidad de parámetro

Desde A a B cuando B es un parámetro de un método de A

Relativamente temporal porque persiste sólo en el alcance de un método

<<P>>

■ Visibilidad local

B es un objeto local en un método de A

Relativamente temporal porque persiste sólo en el alcance de un método

<<L>>

■ Visibilidad global

B es de algún modo visible globalmente

Relativamente persistente porque persiste mientras existan A y B

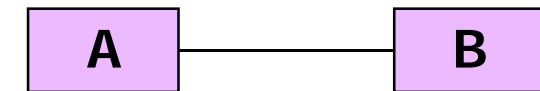
<<G>>

DISEÑO DE LOS CASOS DE USO

A.4 Establecer tipos de enlaces entre objetos: Estereotipo de visibilidad

Visibilidad es la capacidad de un objeto de “ver” o tener una referencia a otro objeto

Para determinar el tipo de visibilidad entre estos dos objetos responder a las siguientes preguntas



¿El objeto de la clase A conoce al objeto de la clase B solo para esta operación?

SI -----> **parámetro o local**

¿El objeto de la clase B ha entrado como parámetro de la operación?



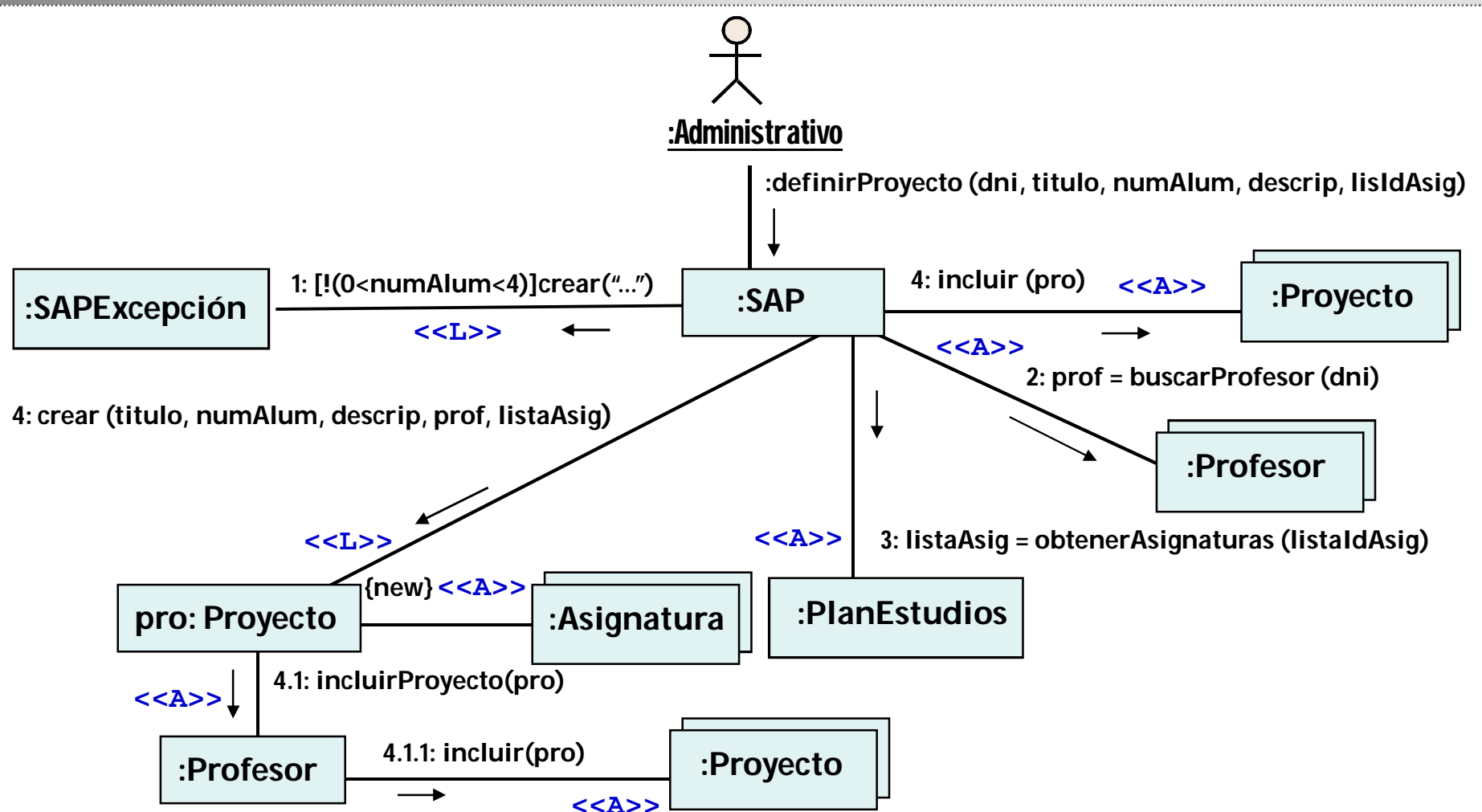
NO -----> **global o asociación**

¿El objeto de la clase B se necesita conocer fuera del ámbito del objeto de la clase A?



DISEÑO DE LOS CASOS DE USO

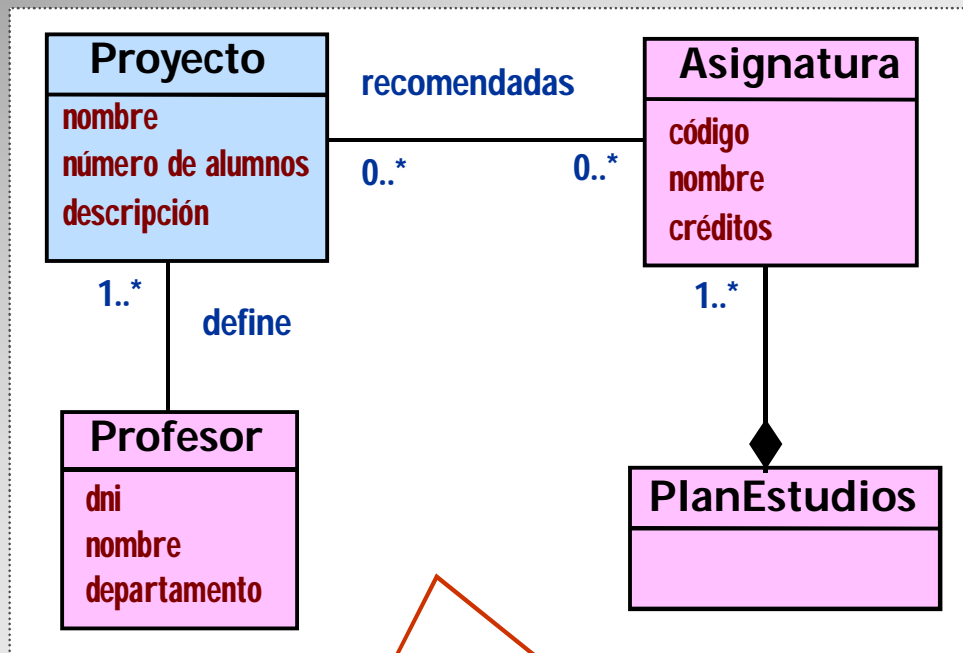
A.4 Establecer tipos de enlaces entre objetos



DISEÑO DE LOS CASOS DE USO

B Inicialización del sistema

- Identificar objetos que se han usado pero no se han creado
- Elaborar un contrato para cada operación que inicialice a esos objetos
- Desarrollar el diagrama de comunicación correspondiente



En **rosa** los objetos que se han usado y no se han inicializado

Operaciones de las que habría que hacer el contrato y el diagrama de comunicación

IncluirPlanEstudios()

Para todos los profesores que imparten proyectos:

IncluirProfesor (dni, nombre, departamento)

Para todas las asignaturas de PlanEstudios:

IncluirAsignatura (código, nombre, créditos)

DISEÑO DE LOS CASOS DE USO

C Establecer relaciones entre el modelo y la Interfaz de Usuario

- Diseñar la Interfaz de Usuario
- Para cada elemento de la interfaz, establecer la comunicación con el modelo

