
Diseño de los casos de uso

Contenido

Modelo de interacción de objetos

Patrones de diseño

Patrones de diseño de Craig Larman

Elaboración del modelo de interacción de objetos

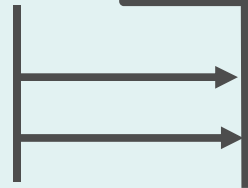
Modelo de interacción de objetos

Modelo de Análisis

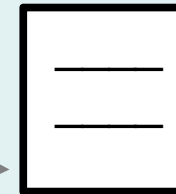
Diagrama conceptual



: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

Patrones de diseño

Problema que resuelven:

Asignar responsabilidades a objetos

Una responsabilidad es una obligación que debe tener un objeto en su comportamiento, y debe incluirse en algún método u operación

Las responsabilidades de un objeto pueden ser:

+ Conocer

- Los datos privados encapsulados por él
- Los objetos relacionados con él
- La información que puede calcular o derivar

+ Hacer

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

Responsabilidad \neq Método► Cumplen las responsabilidades

└───► Se implementan usando métodos que operan solos o en colaboración con otros métodos

Patrones de diseño

¿Qué es un patrón de diseño?

Descripción de un problema con su solución en un determinado contexto

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

Enunciado de la solución que se propone para un contexto determinado

Consecuencias

Buenas y malas que pueden ocasionar el uso del patrón

Ayudan a entender si es factible usar el patrón en una situación particular

Patrones de diseño de Craig Larman

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
- Negociar una solución razonable
- Son un ejemplo de fuerza de abstracción porque dan nombre a una idea compleja

Patrones Craig Larman o patrones GRASP

- Experto en información
- Creador
- Bajo acoplamiento
- Alta cohesión
- Controlador
- Polimorfismo, fabricación pura, indirección, no hables con extraños, ...

Patrones de diseño de Craig Larman

Patrón experto en información

Nombre: Experto en información

Problema

Complejidad en la búsqueda de información y acoplamientos fuertes entre clases en estas búsquedas

Solución

Asignar responsabilidad a la clase que contiene la información necesaria para llevar a cabo dicha responsabilidad

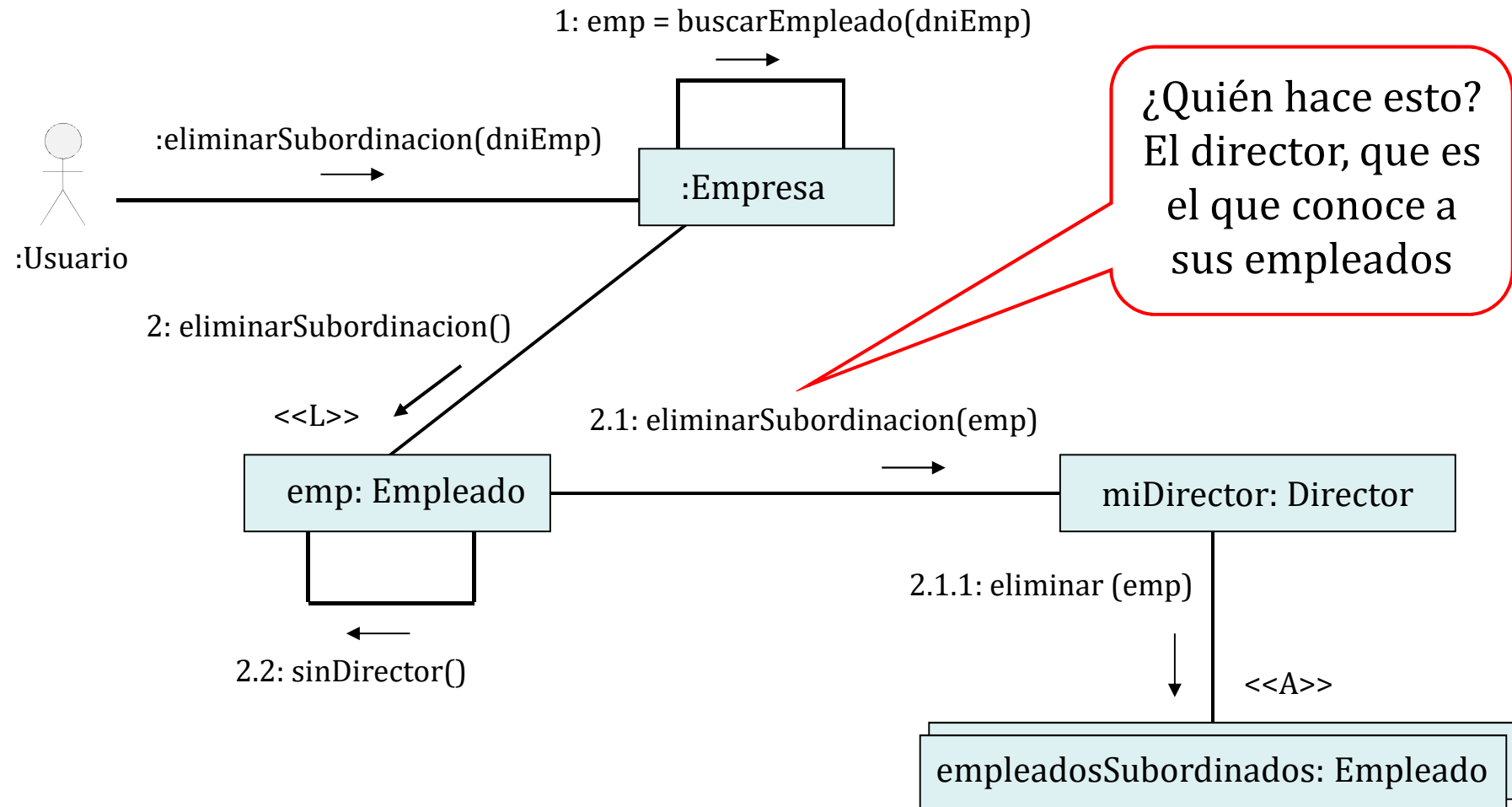
Consecuencias

Malas: en ocasiones va en contra de los principios de acoplamiento o cohesión

Buenas: mantiene el ocultamiento de la información y distribuye el comportamiento

Patrones de diseño de Craig Larman

Ejemplo de patrón experto en información



Patrones de diseño de Craig Larman

Patrón creador

Nombre: Creador

Problema

Tener acoplamientos, mala encapsulación y reutilización y poca claridad en la construcción de objetos

Solución

Asignar a la clase B la responsabilidad de crear una instancia 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

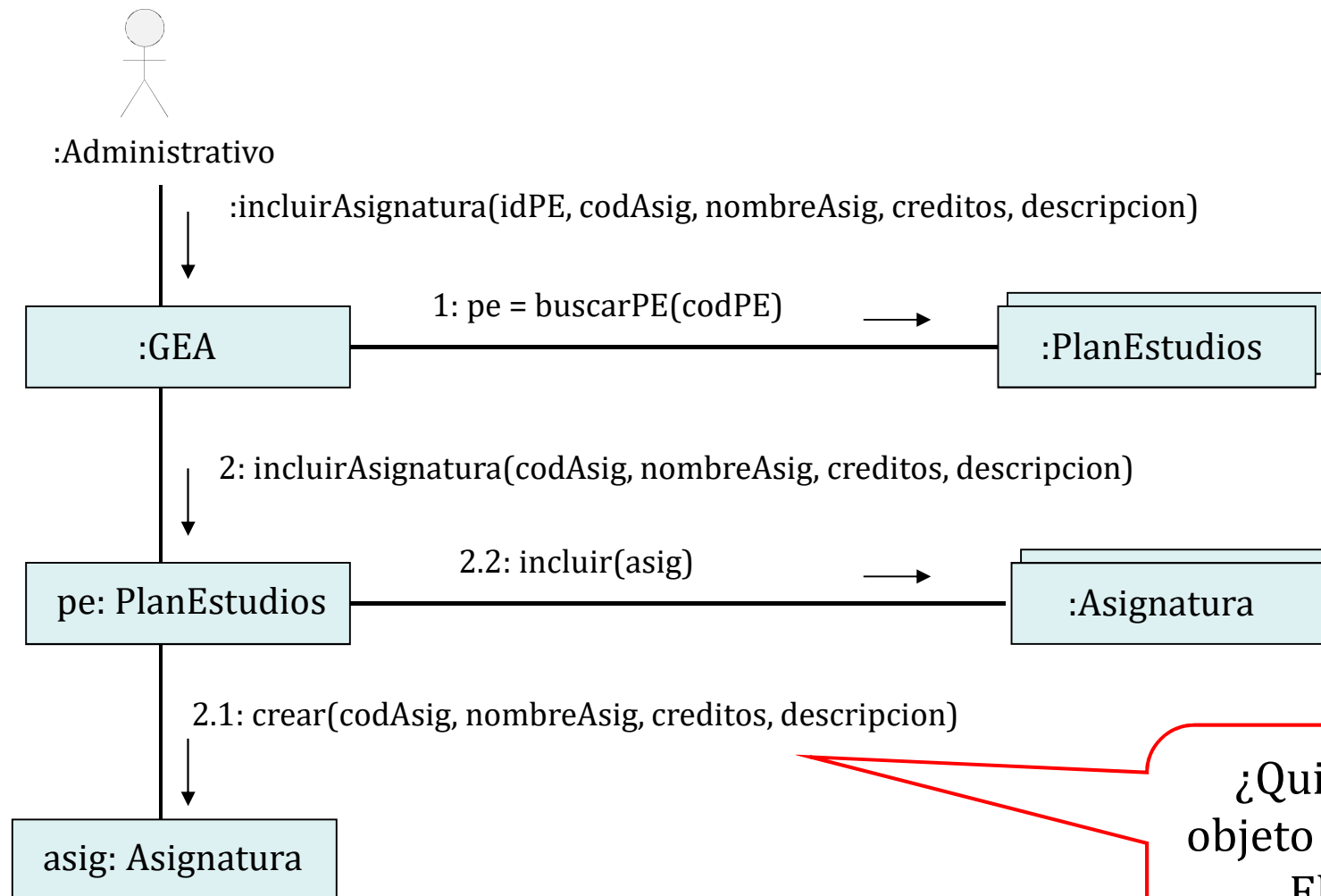
Consecuencias

Malas: no es conveniente su uso cuando se construye a partir de instancias existentes

Buenas: produce bajo acoplamiento

Patrones de diseño de Craig Larman

Ejemplo de patrón creador



¿Quién crea el
objeto Asignatura?
El que lo
conoce/agrega

Patrones de diseño de Craig Larman

Patrón bajo acoplamiento

Nombre: Bajo acoplamiento

Problema

Elementos que dependen de demasiados elementos. Una modificación conlleva demasiadas modificaciones colaterales, difíciles de entender aisladamente y difíciles de reutilizar

Solución

Asignar responsabilidades de forma que se tengan elementos (clases, subsistemas, ...) que dependan solamente de los únicos elementos que necesite

Consecuencias

Malas: Llevada al 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 elementos
Fáciles de entender de manera aislada
Aumento de la reutilización

Patrones de diseño de Craig Larman

Patrón alta cohesión

Nombre: Alta cohesión

Problema

Elementos con pocas tareas o con muchas pero no relacionadas. Estos elementos son difíciles de entender, de reutilizar y de mantener, además se ven afectados por continuos cambios

Solución

Asignar responsabilidades de forma que todas las tareas de un elemento (clase, subsistema, ..) estén para lograr un objetivo común

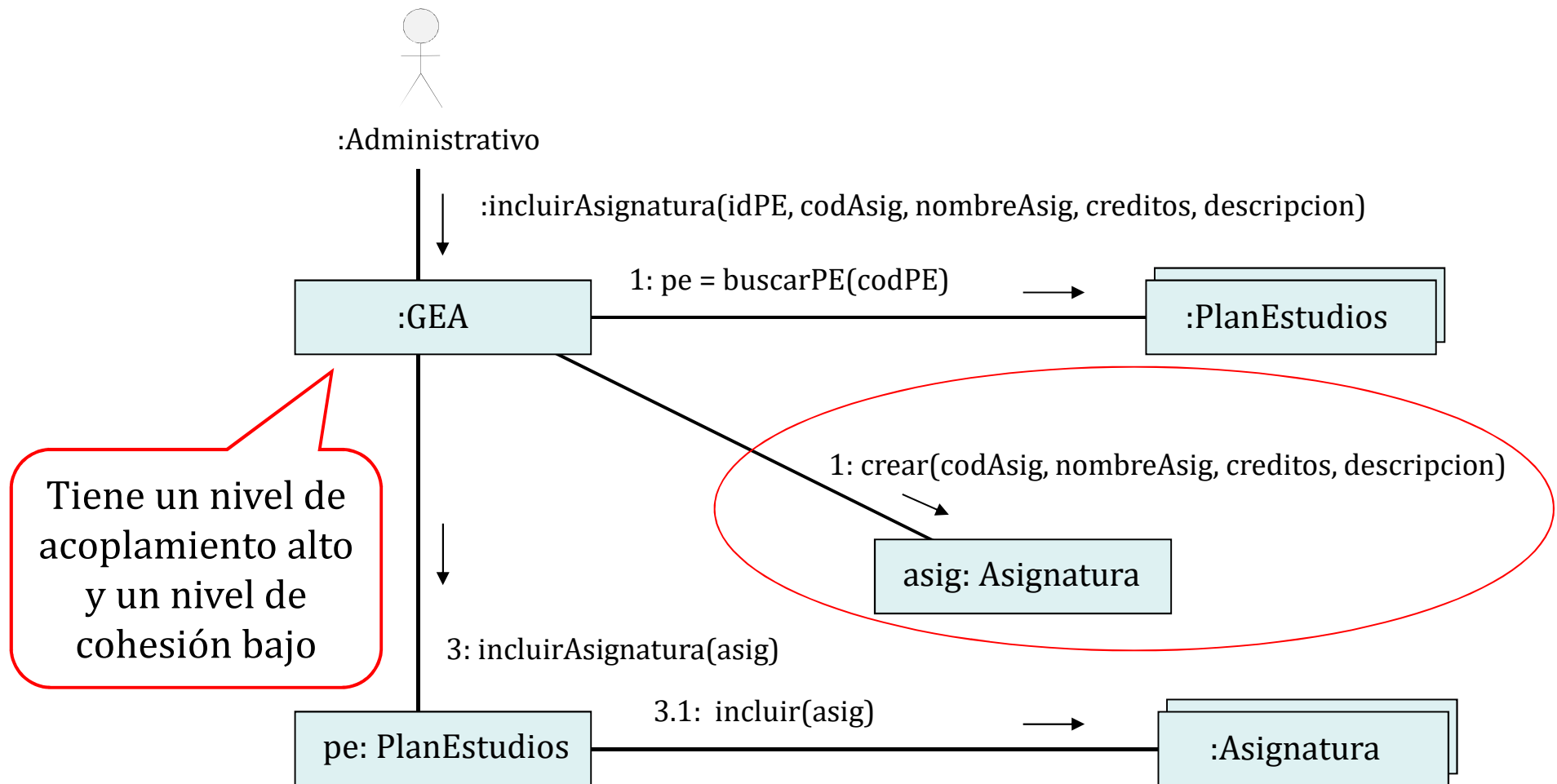
Consecuencias

Malas: Ninguna, renunciar a la alta cohesión tan solo cuando esté muy justificado

Buenas: Claridad y facilidad de entendimiento del diseño
Simplificación del mantenimiento y de las mejoras
Aumento de la reutilización

Patrones de diseño de Craig Larman

Ejemplo de patrón bajo acoplamiento y alta cohesión



Patrones de diseño de Craig Larman

Patrón controlador o fachada

Nombre: Controlador o fachada

Problema

Comunicación entre los objetos de la capa del dominio de la solución y la capa de la interfaz

Solución

Asignar responsabilidades 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

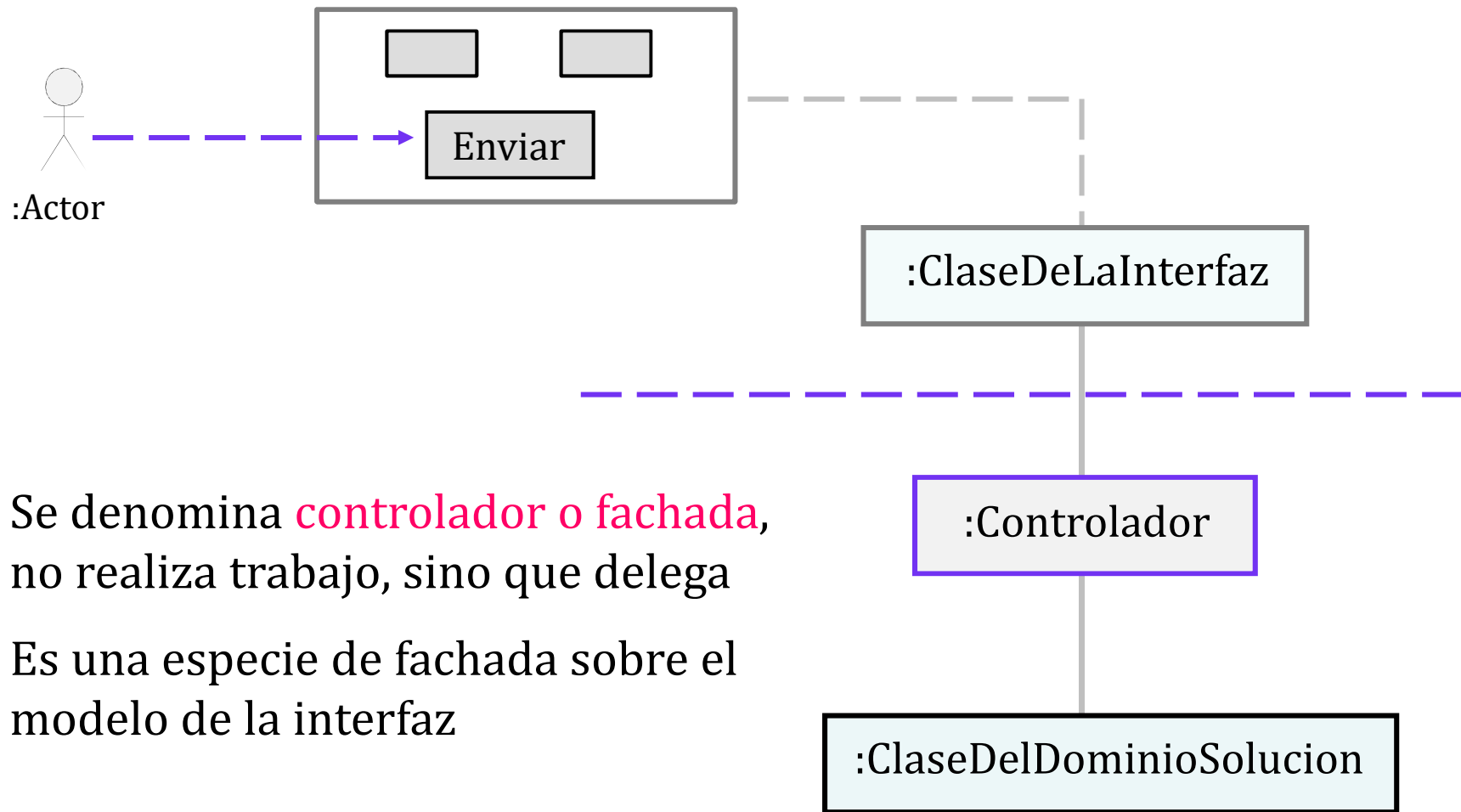
Buenas: Asegura que la lógica de la aplicación no se maneja en la interfaz

Buena reutilización y bajo nivel de acoplamiento

Posibilidad de razonar sobre el estado de los casos de uso

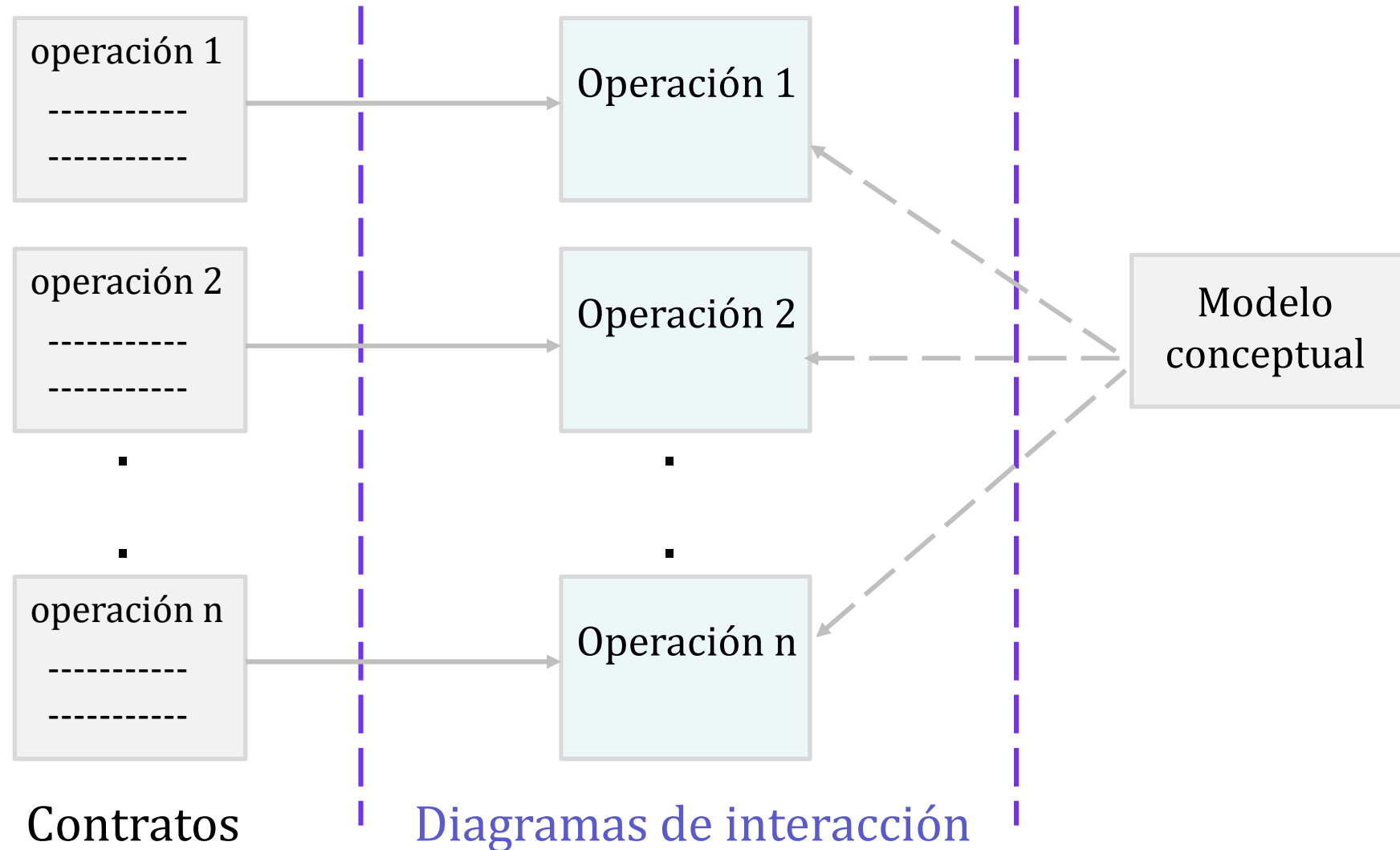
Patrones de diseño de Craig Larman

Ejemplo de patrón controlador o fachada



Elaboración del modelo de interacción

De dónde partimos y hacia dónde vamos



Elaboración del modelo de interacción

Directrices generales

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

Elaboración del modelo de interacción

Pasos a seguir

A. Elaborar los diagramas de interacción, de forma que para cada operación especificada en los DSS:

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

B. Inicialización del sistema

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

Elaboración del modelo de interacción

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

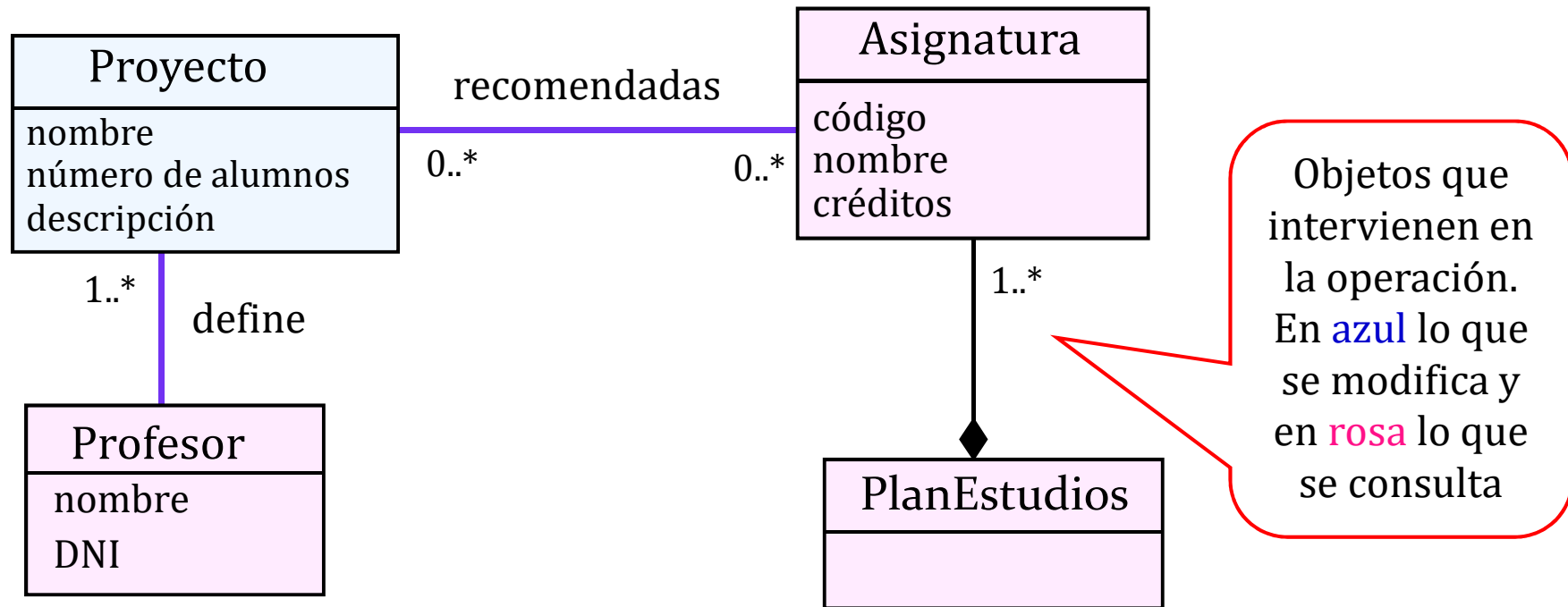
Básico: El contrato de la operación

Nombre	definirProyecto(idProf, titulo, numAlum, descripcion, listaIdAsig)
Responsabilidad	Añade un nuevo proyecto a la lista de proyectos definidos por el profesor y establece sus asignaturas recomendadas del plan de estudios
Tipo	SAP
Notas	
Excepciones	<ul style="list-style-type: none">- Si el profesor identificado por idProf no existe- Si $!(0 < \text{numAlum} < 4)$- Si alguna de las asignaturas identificadas por alguno de los elementos de listaIdAsig no existe
Salida	
Precondiciones	
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 idProf Para cada elemento de listaIdAsig: <ul style="list-style-type: none">- Fue creado un enlace entre pro y el objeto de la clase Asignatura identificado por el correspondiente elemento de listaIdAsig

Elaboración del modelo de interacción

A.1 (Continuación)

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

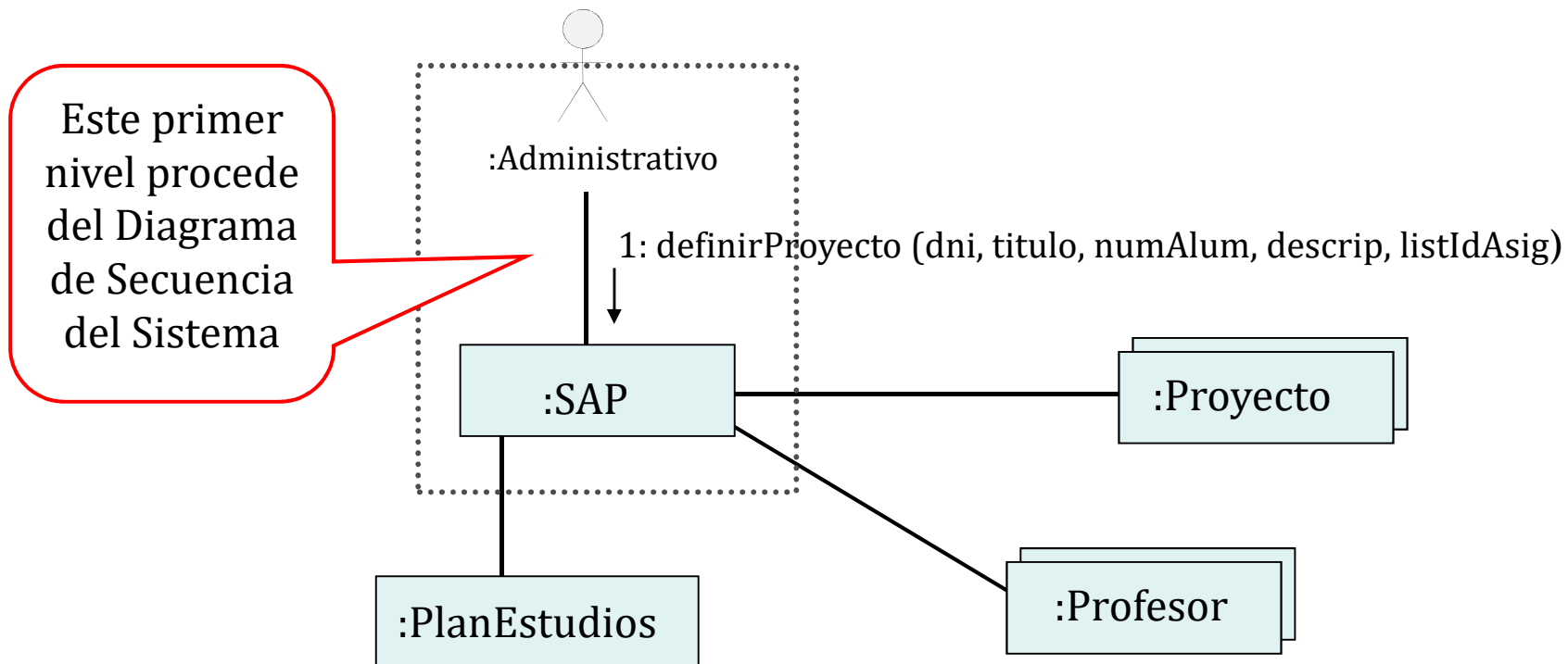


Elaboración del modelo de interacción

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

¿Qué objetos necesita conocer directamente el controlador?

En el ejemplo, necesita conocer directamente a todos los profesores, a todos los proyectos y al plan de estudios



Elaboración del modelo de interacción

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 el objeto en cuestión?

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

Se comienza por el **controlador**

En el ejemplo, el objeto SAP es responsable de:

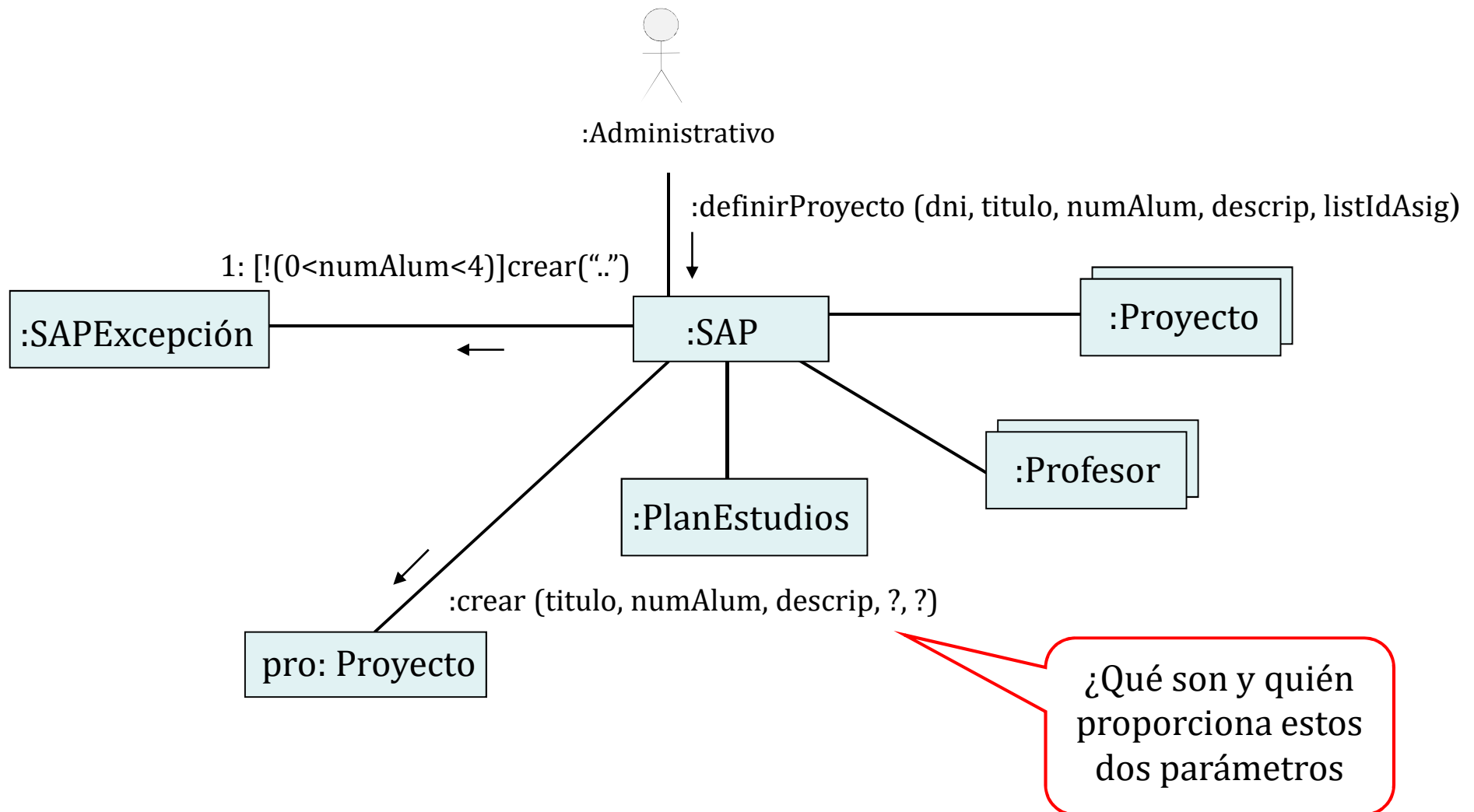
- Cumplir las dos primeras excepciones (experto en información)
- Cumplir con la primera poscondición (creador)

Simplificación:

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

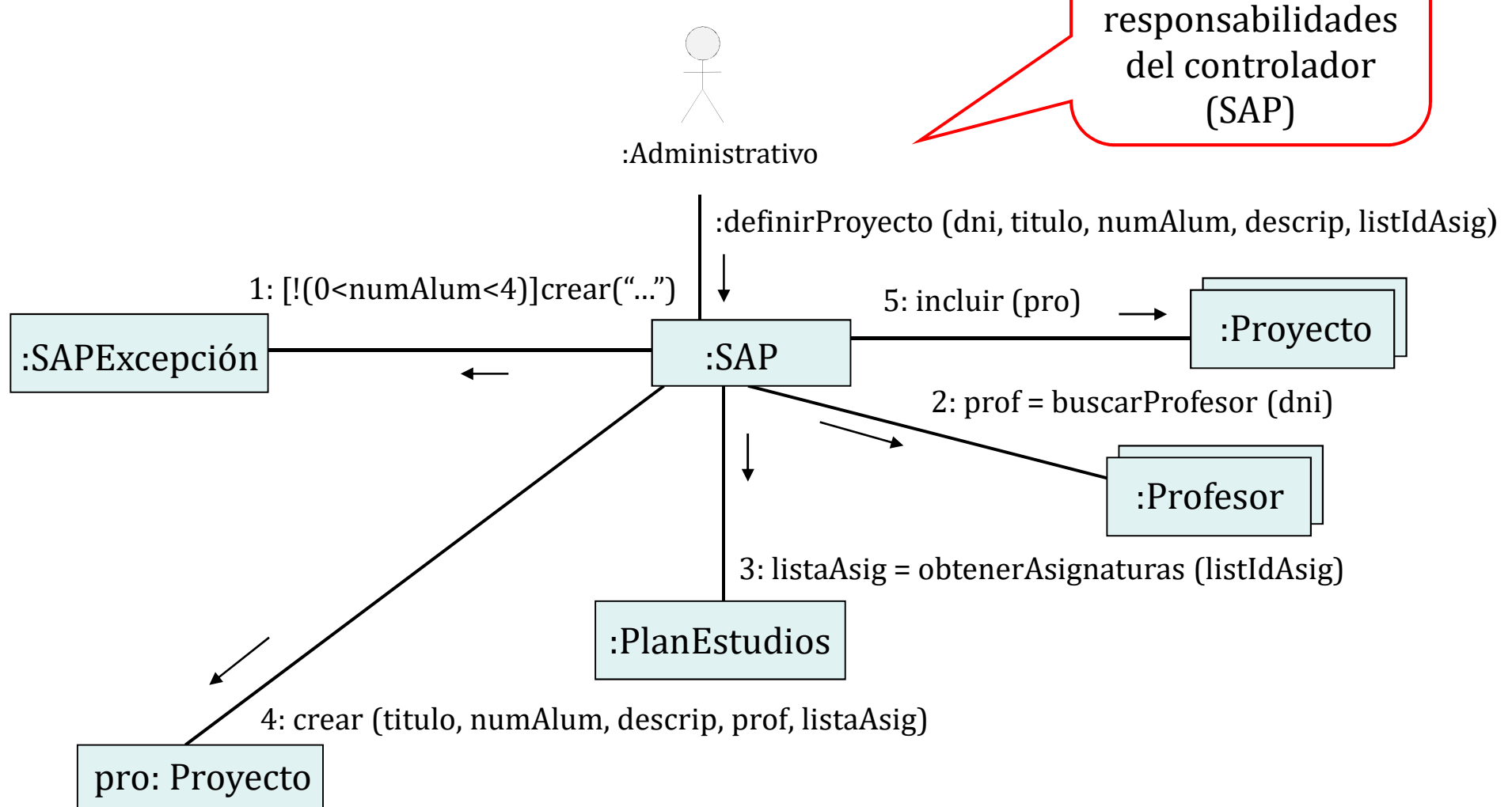
Elaboración del modelo de interacción

A.3: (Continuación)



Elaboración del modelo de interacción

A.3: (Continuación)



Elaboración del modelo de interacción

A.3: (Continuación)

Seguir asignando responsabilidades a los objetos que van apareciendo hasta cumplir con el contrato, haciendo siempre la pregunta

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

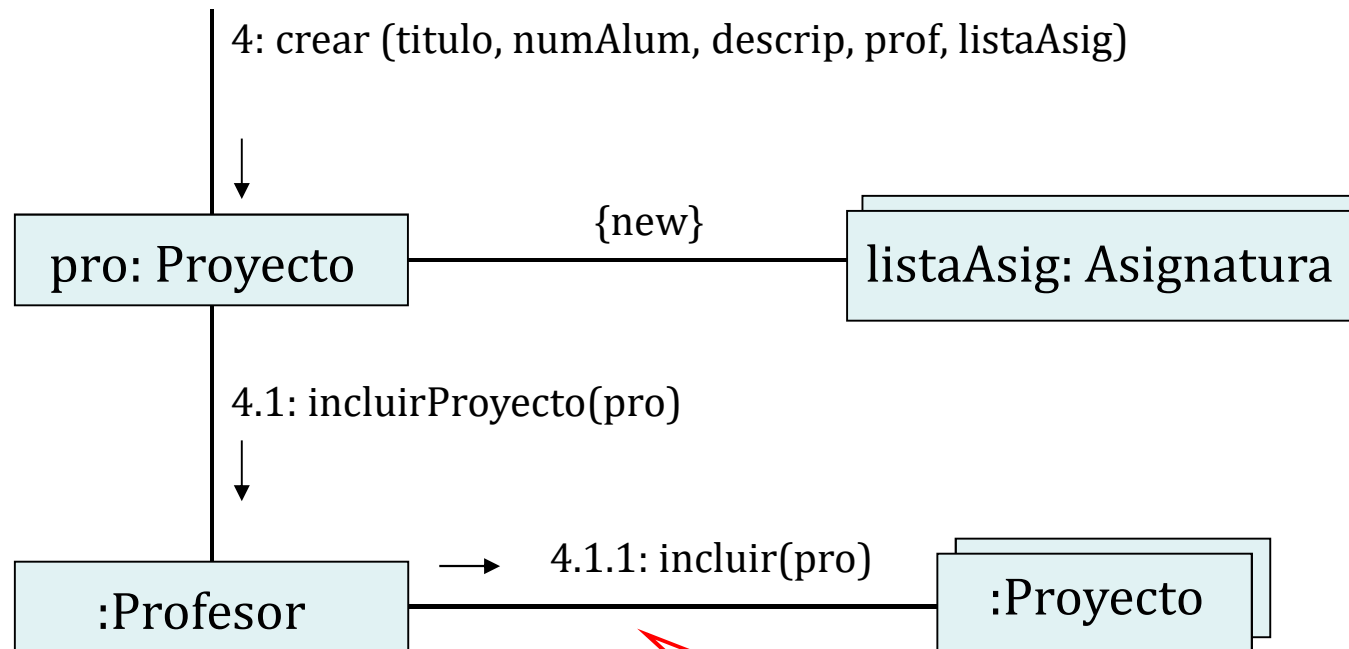
Al igual que antes, la respuesta es aplicar los patrones de diseño, fundamentalmente, el experto en información y el creador

En el ejemplo, **pro**, el proyecto recién creado es responsable de:

- Cumplir con las dos últimas poscondiciones (experto en información), es decir, construir los enlaces correspondientes

Elaboración del modelo de interacción

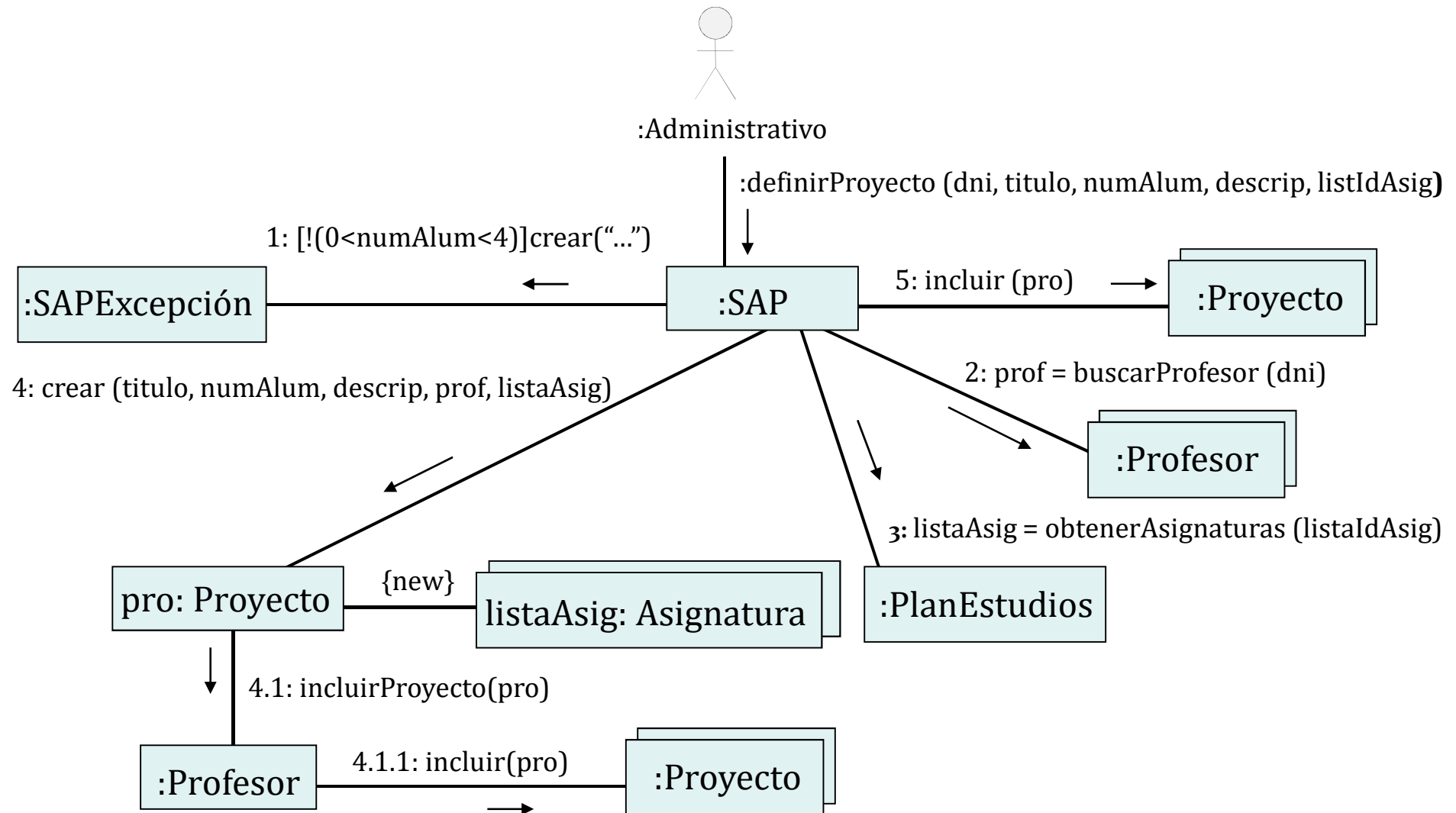
A.3: (Continuación)



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

Elaboración del modelo de interacción

A.3: (Continuación)



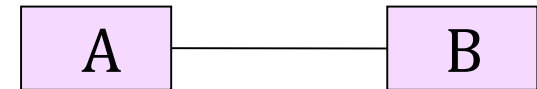
Elaboración del modelo de interacción

A.4: Establecer tipos de enlaces entre objetos

(estereotipos de visibilidad)

Visibilidad: 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?

SI -----> parámetro <<P>>

NO -----> local <<L>>

NO -----> global o asociación

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

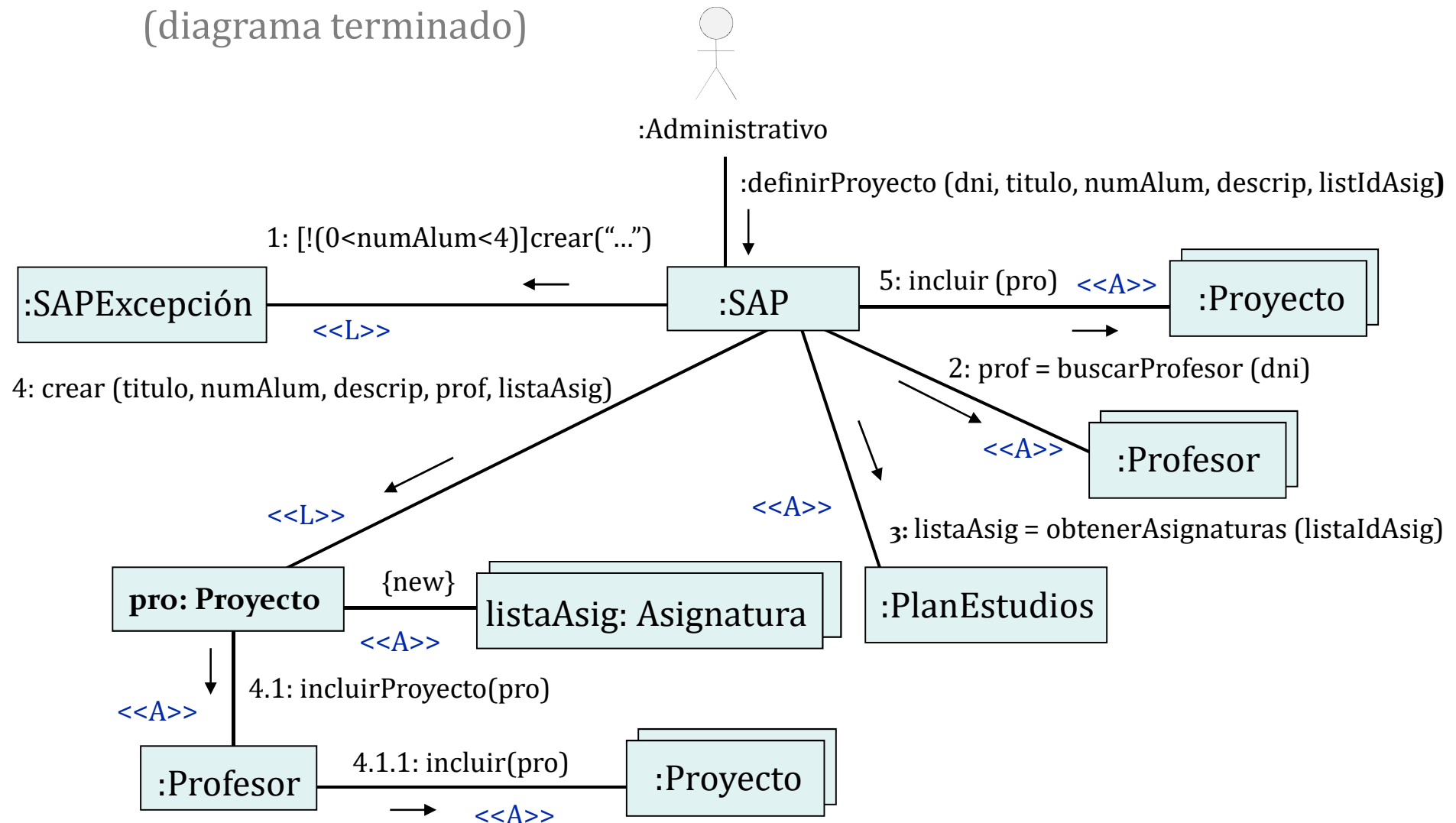
SI -----> global <<G>>

NO -----> asociación <<A>>

Elaboración del modelo de interacción

A.4: Establecer tipos de enlaces entre objetos

(diagrama terminado)



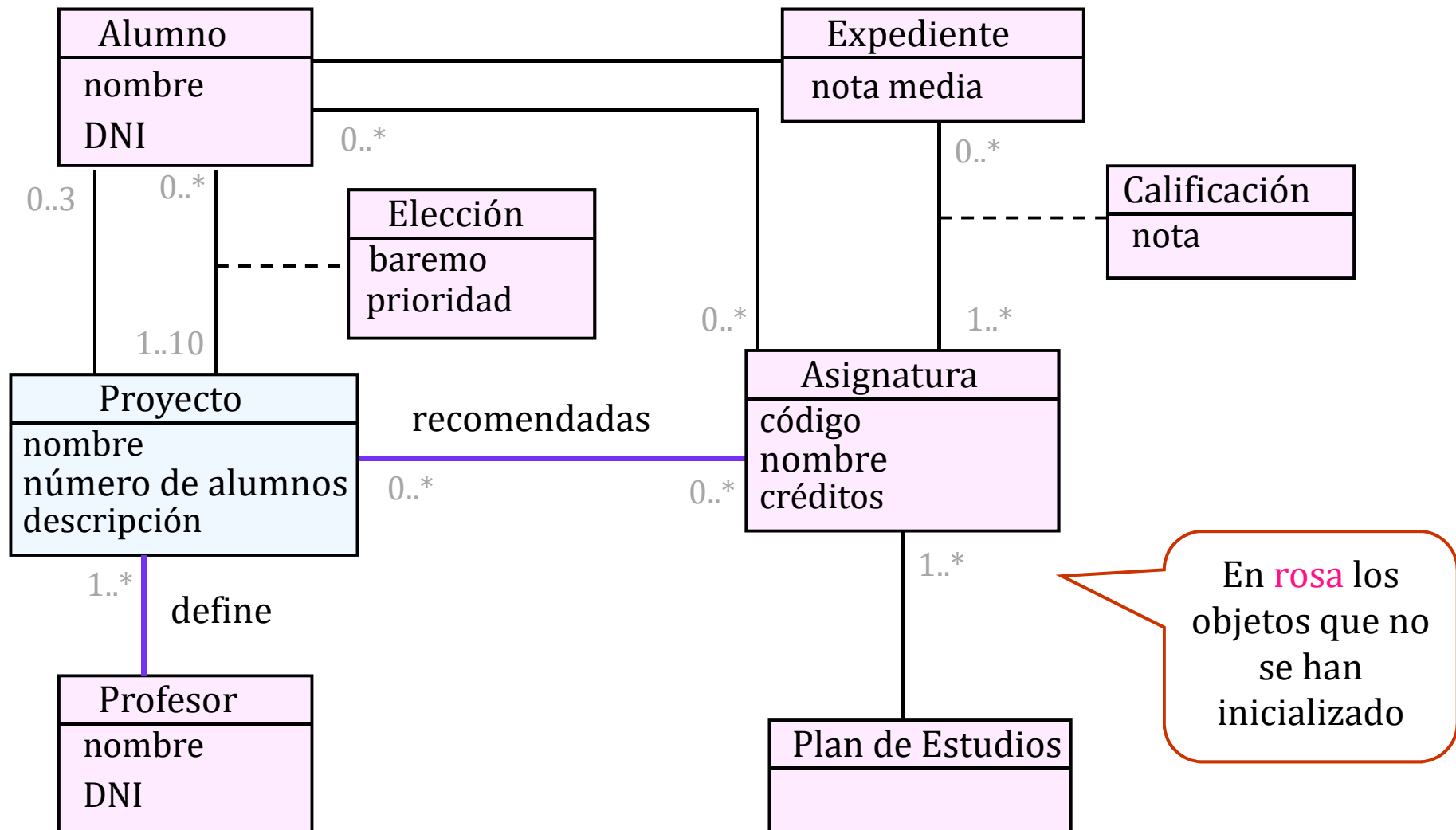
Elaboración del modelo de interacción

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

Elaboración del modelo de interacción

B: Inicialización del sistema (Continuación)



Elaboración del modelo de interacción

B: Inicialización del sistema (Continuación)

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 del Plan de Estudios

- IncluirAsignatura (codAsig, nombre, creditos)

Para todos los alumnos:

- IncluirAlumno (dni, nombre)

Para todas las asignaturas del expediente del alumno:

- IncluirNotaAsignatura (dniAlu, codAsig, nota)

Elaboración del modelo de interacción

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 entre este y el modelo

