
Fundamentos de Ingeniería del Software

Apuntes Control 2

Ismael Sallami Moreno

Mayo de 2025

Índice general

Diseño de los casos de uso

Contenido

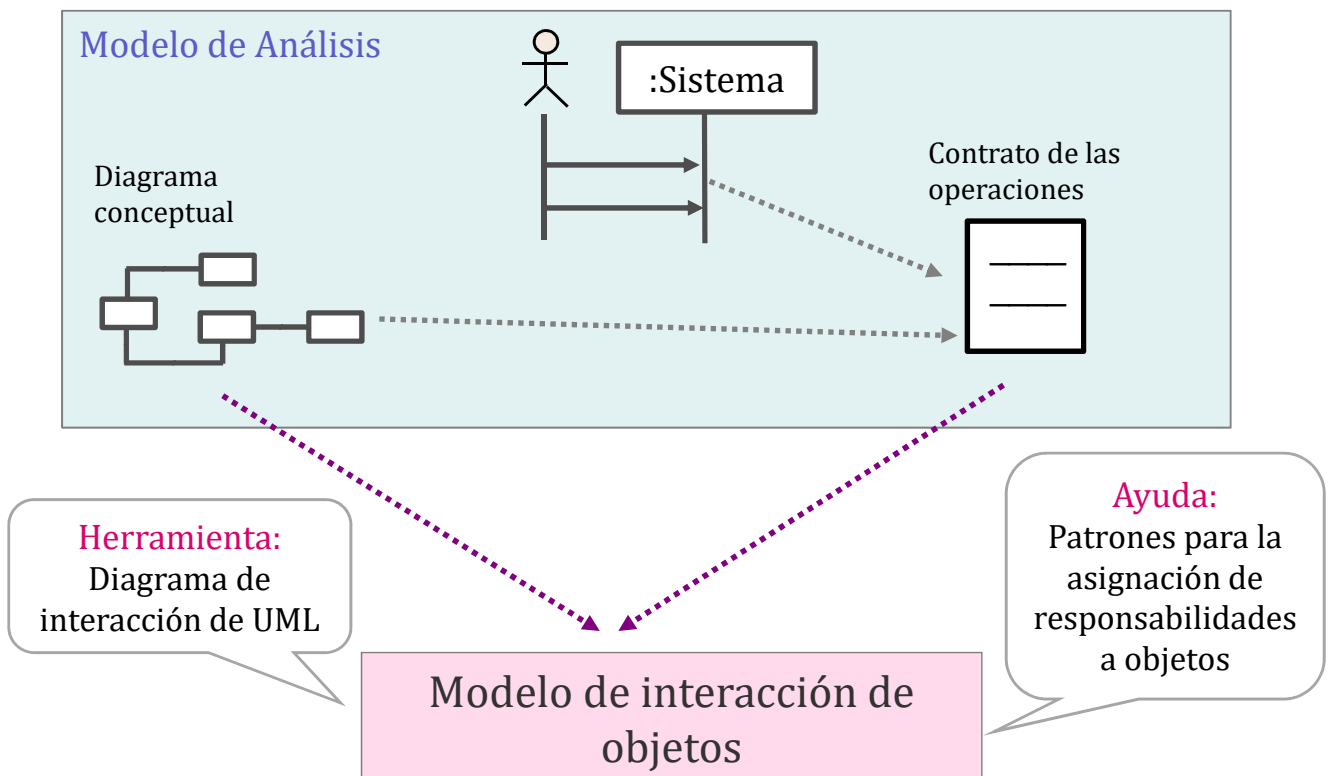
Modelo de interacción de objetos

Patrones de diseño

Patrones de diseño de Craig Larman

Elaboración del modelo de interacción de objetos

Modelo de interacción de objetos



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

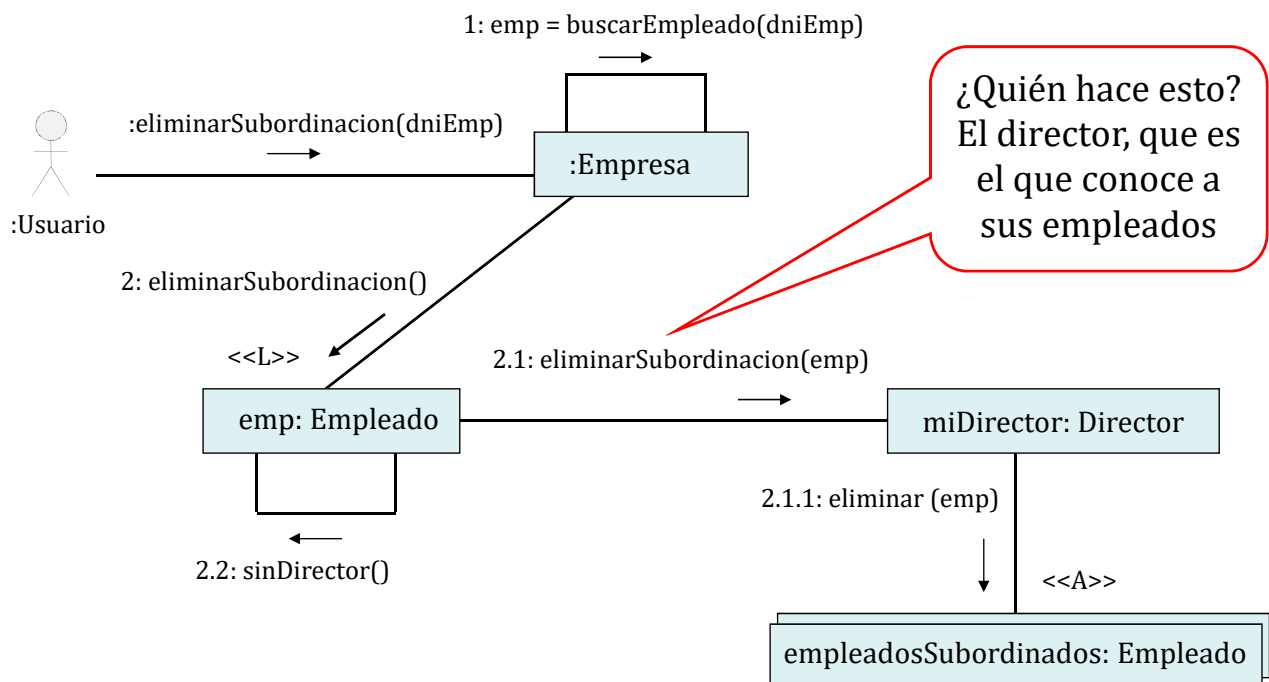
Consecuencias

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

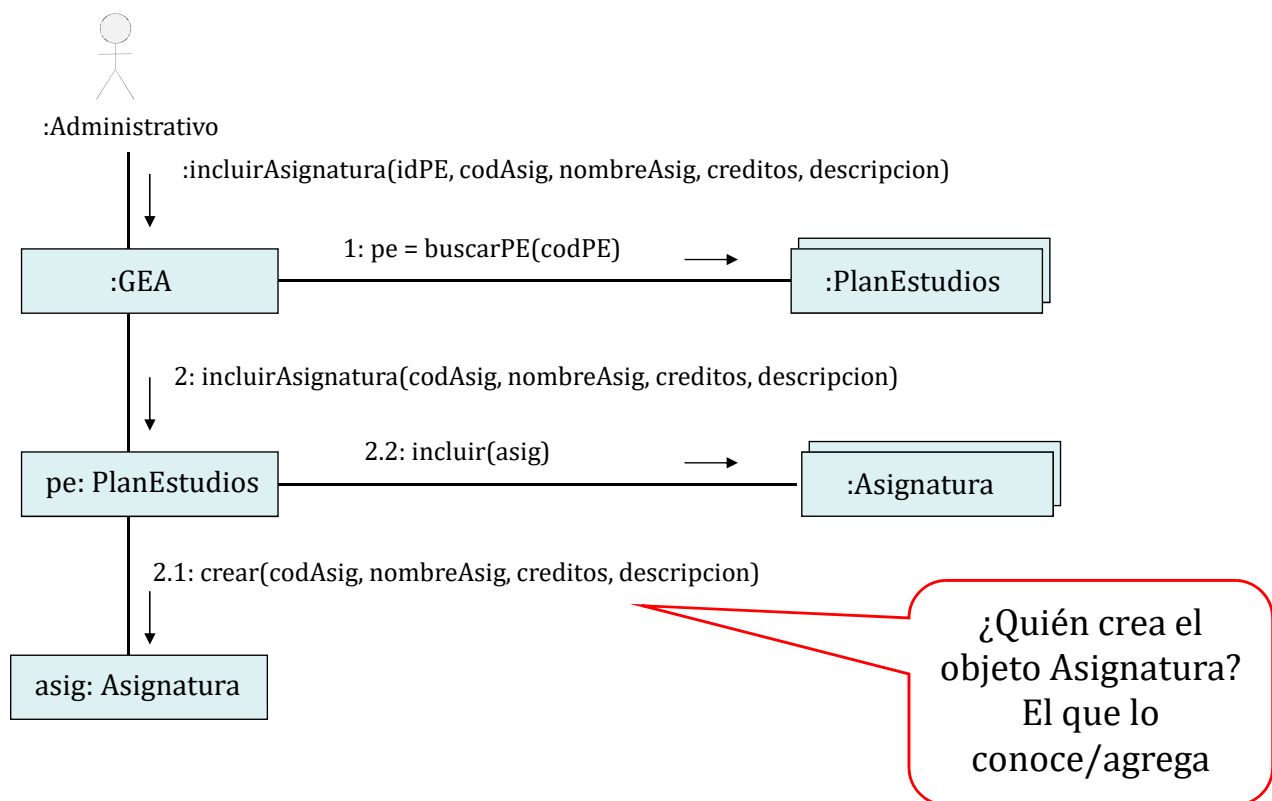
Consecuencias

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



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

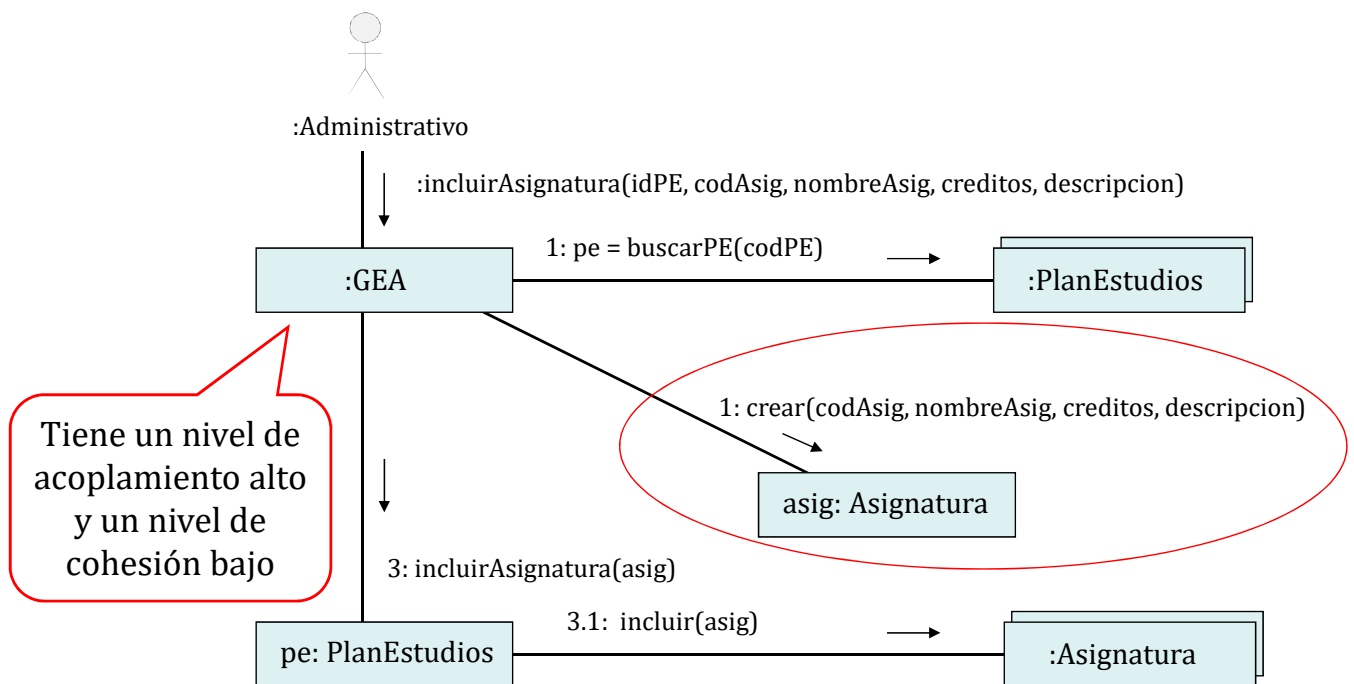
Consecuencias

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)

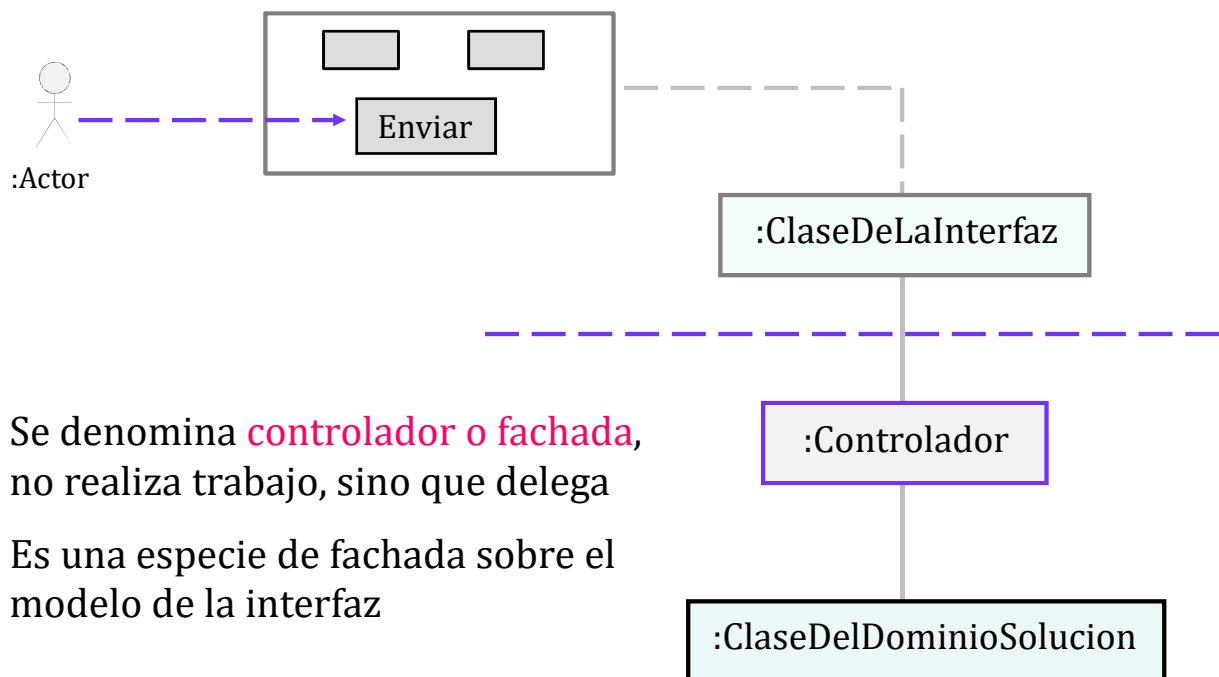
Consecuencias

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

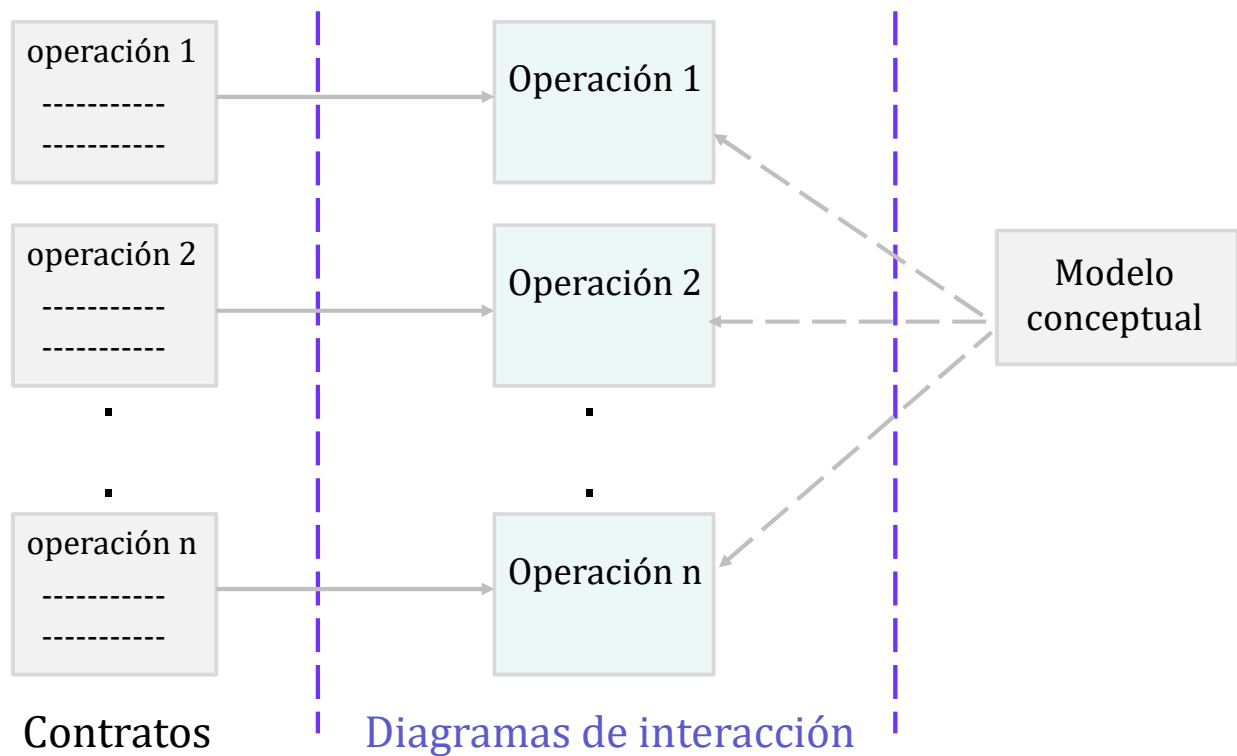
Ejemplo de patrón controlador o fachada



- Se denomina **controlador o fachada**, no realiza trabajo, sino que delega
- Es una especie de fachada sobre el modelo de la interfaz

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

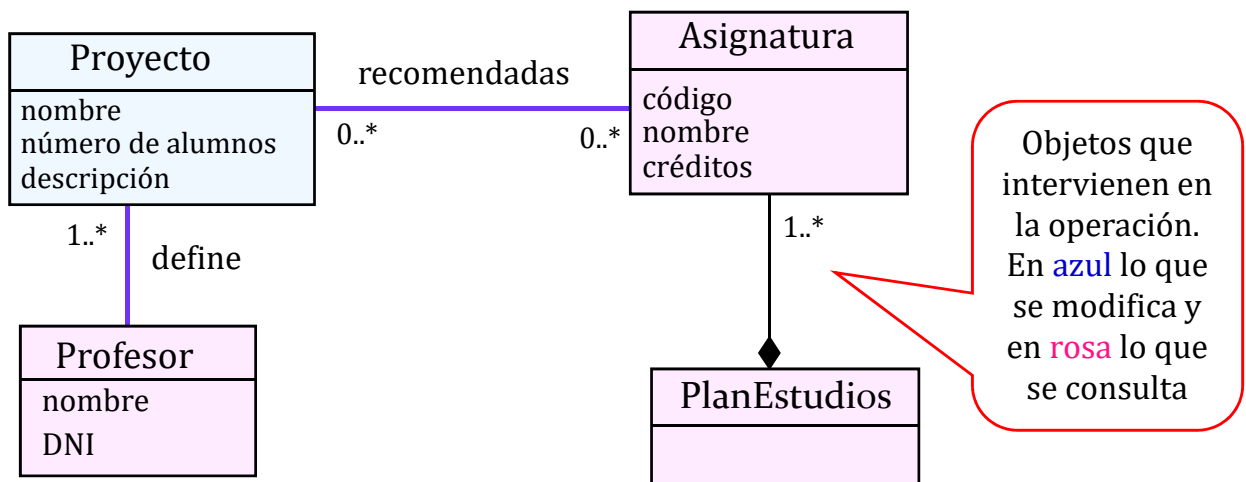
Básico: El contrato de la operación

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

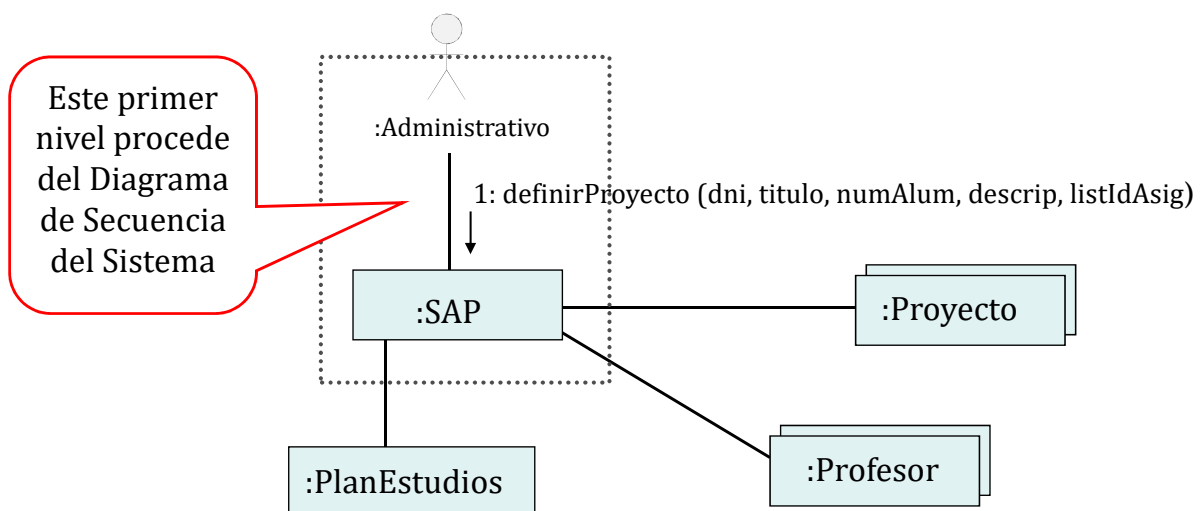


Elaboración del modelo de interacción

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:

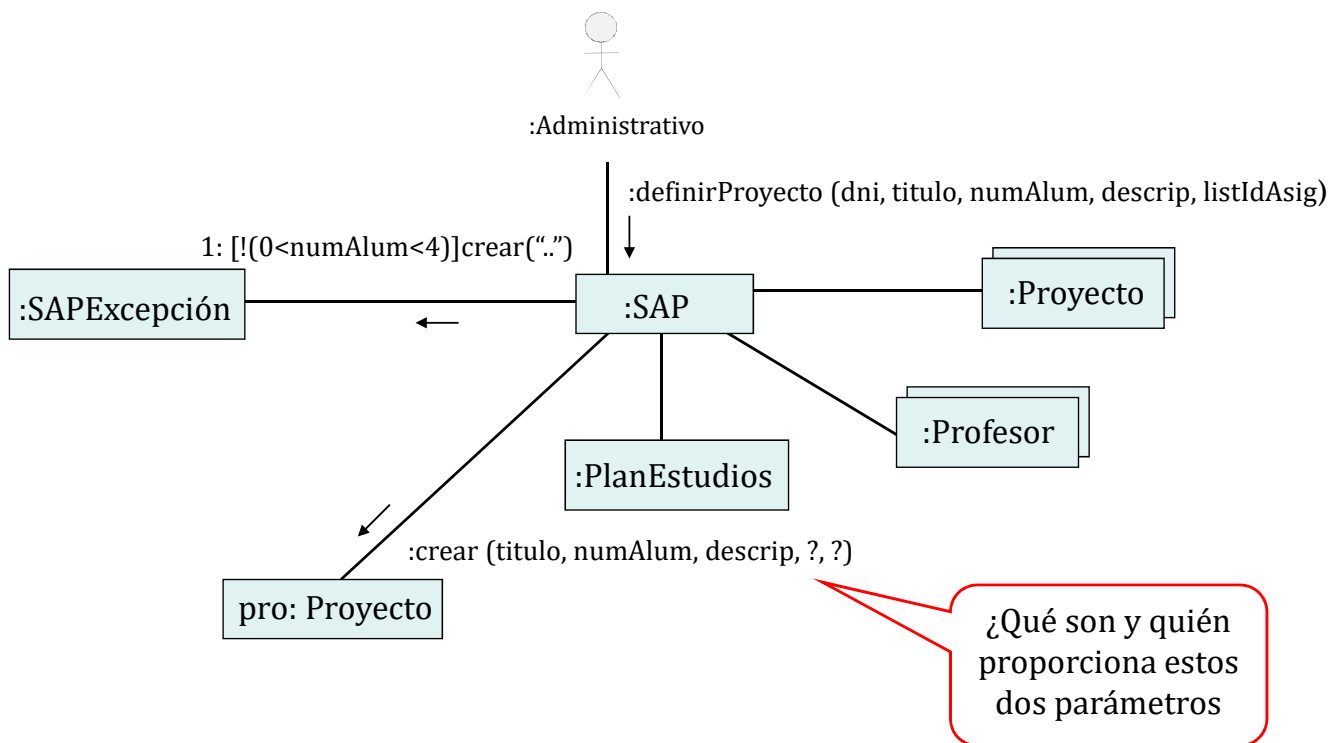
- **Cumplir las dos primeras excepciones** (experto en información)
- **Cumplir con la primera poscondición** (creador)

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

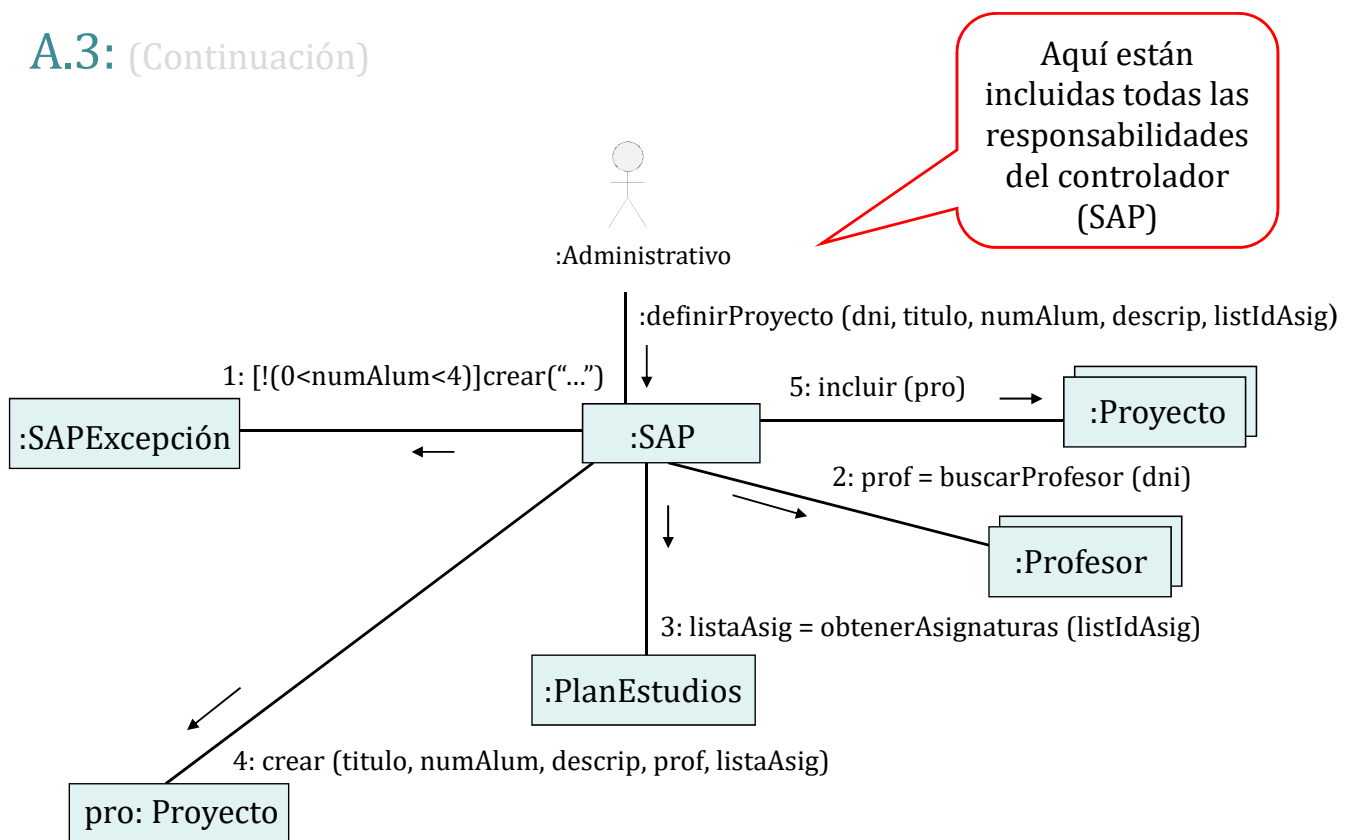
Elaboración del modelo de interacción

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?

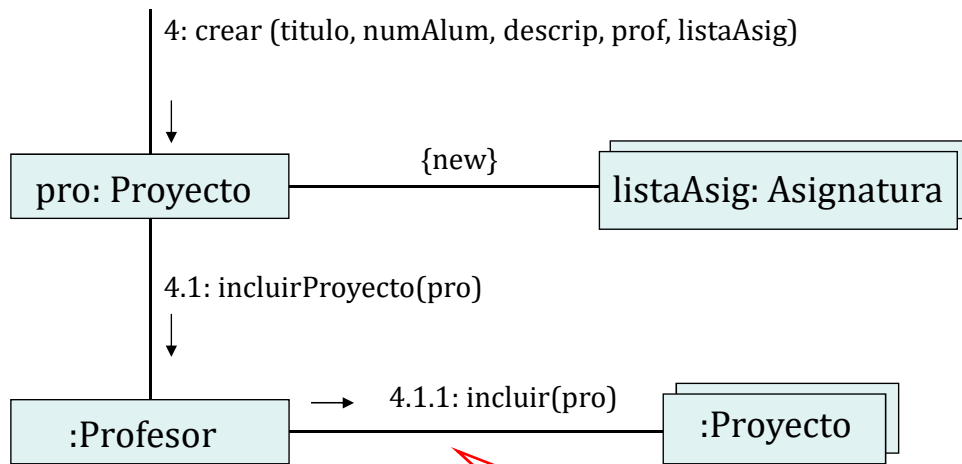
Al igual que antes, la respuesta es aplicar los patrones de diseño, fundamentalmente, el **experto en información** y el **creador**

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

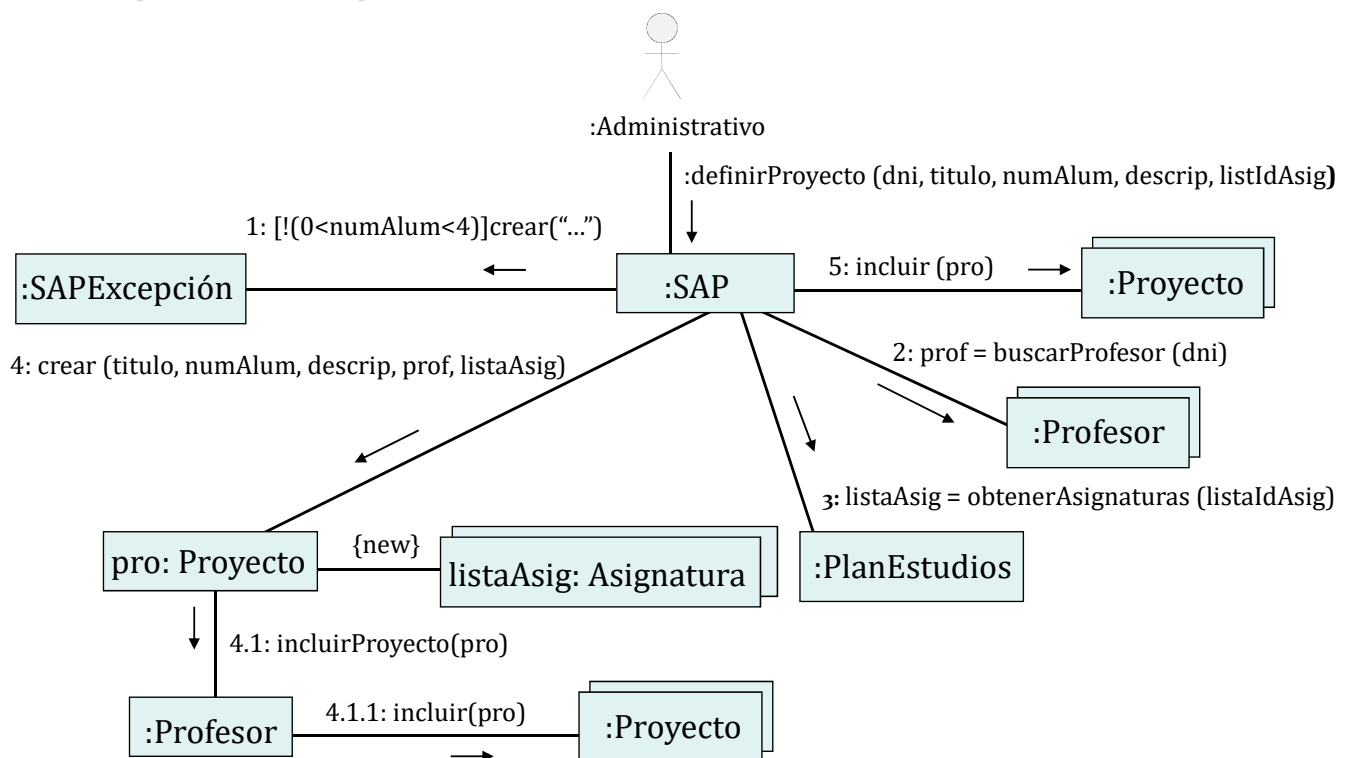
A.3: (Continuación)



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?

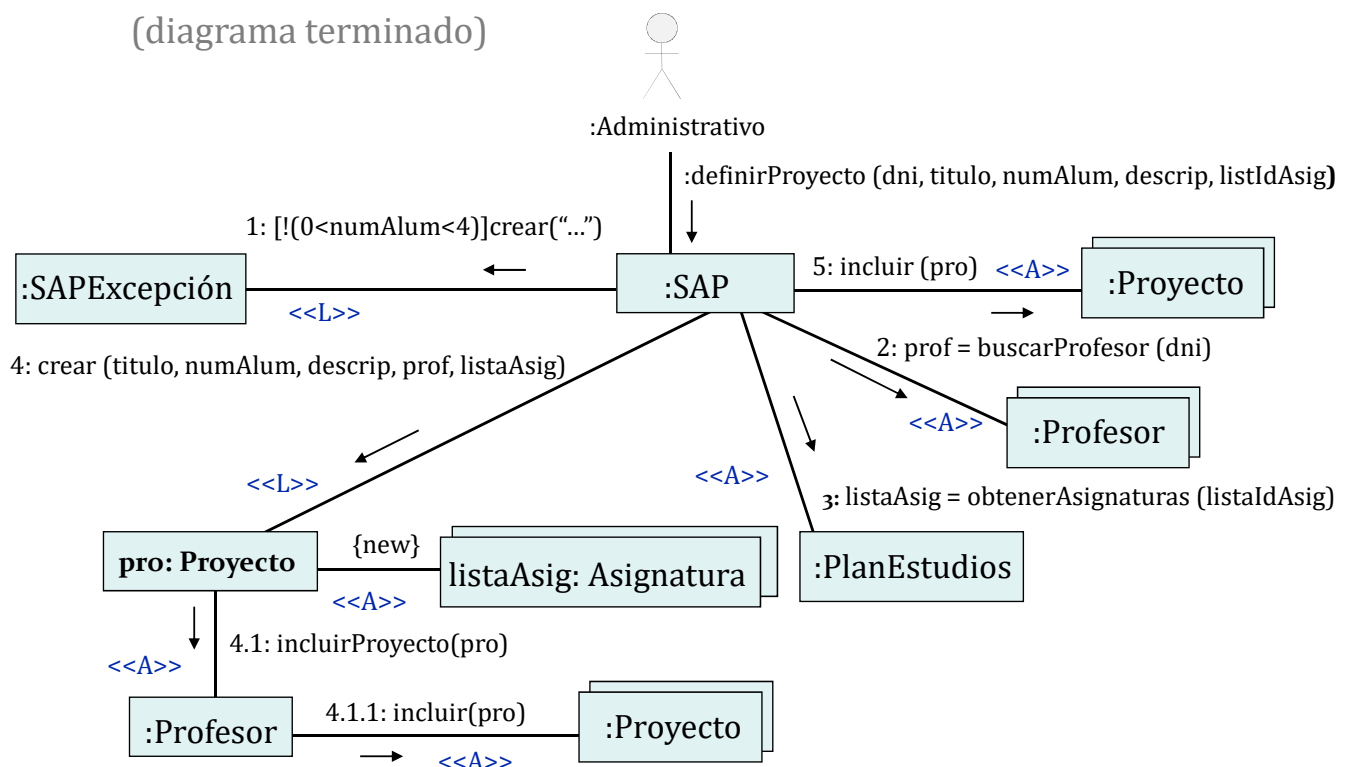
SI -----> global <<G>>

NO -----> asociación <<A>>

Elaboración del modelo de interacción

A.4: Establecer tipos de enlaces entre objetos

(diagrama terminado)



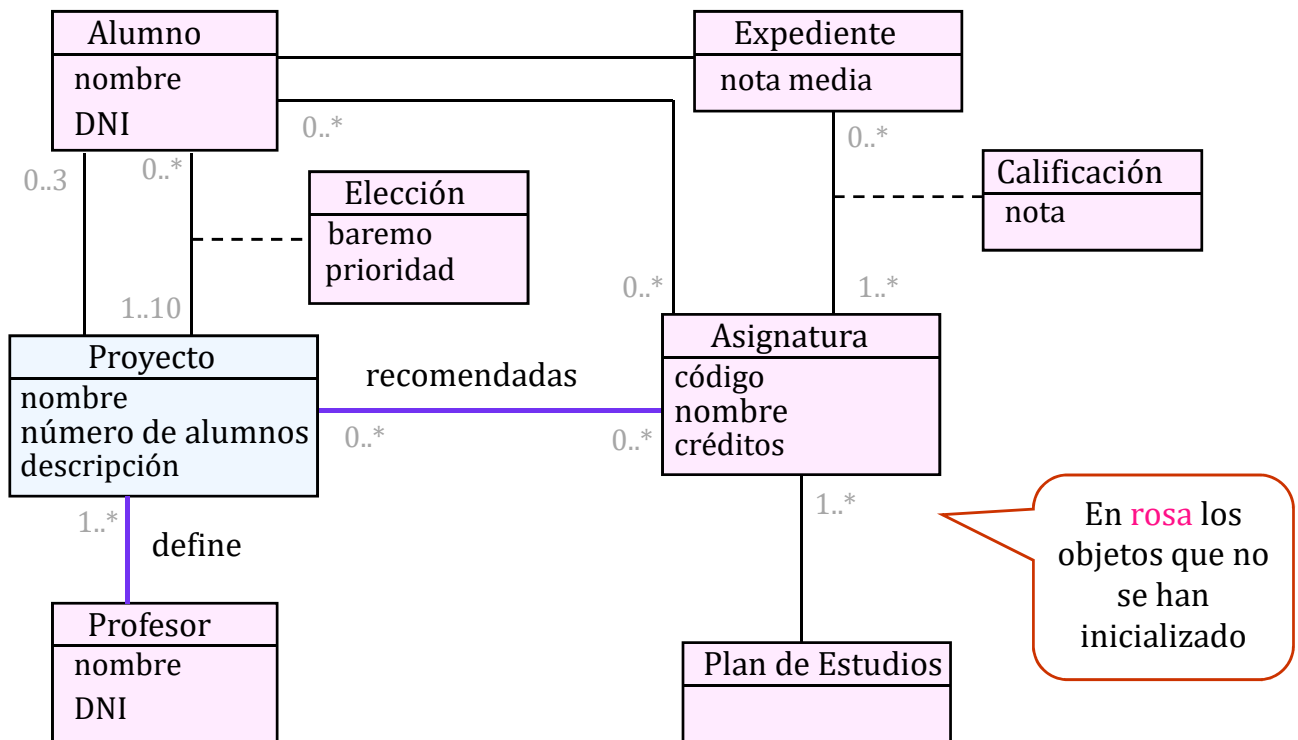
Elaboración del modelo de interacción

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, credits)

Para todos los alumnos:

- IncluirAlumno (dni, nombre)

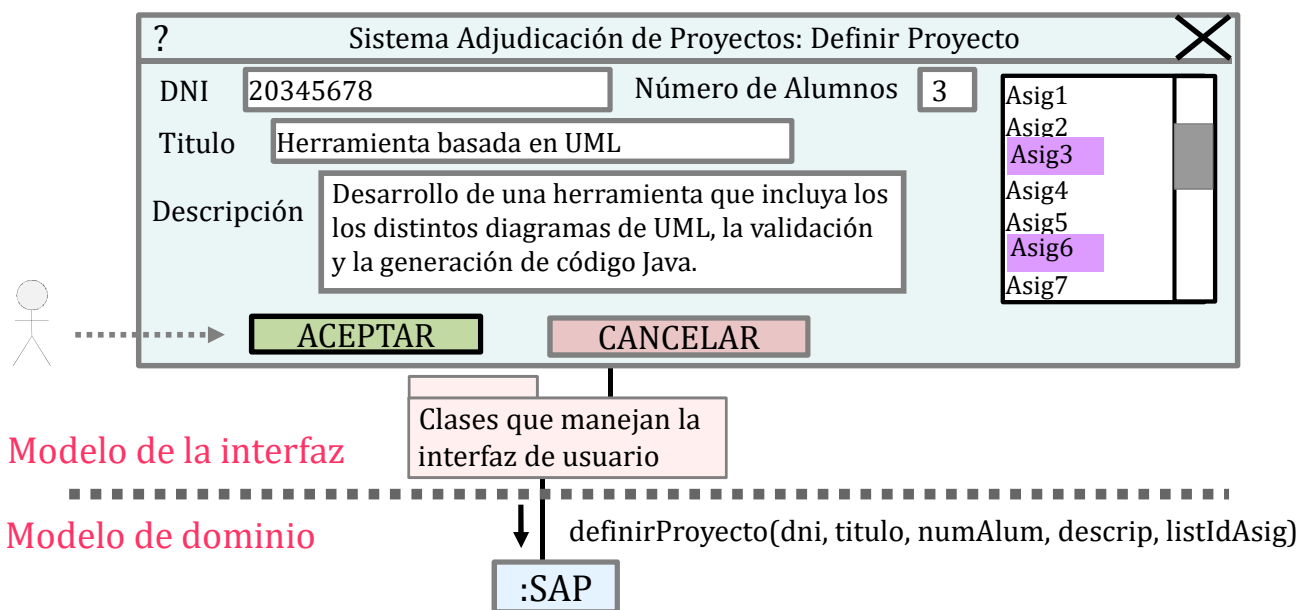
Para todas las asignaturas del expediente del alumno:

- 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



Diseño de la estructura de objetos

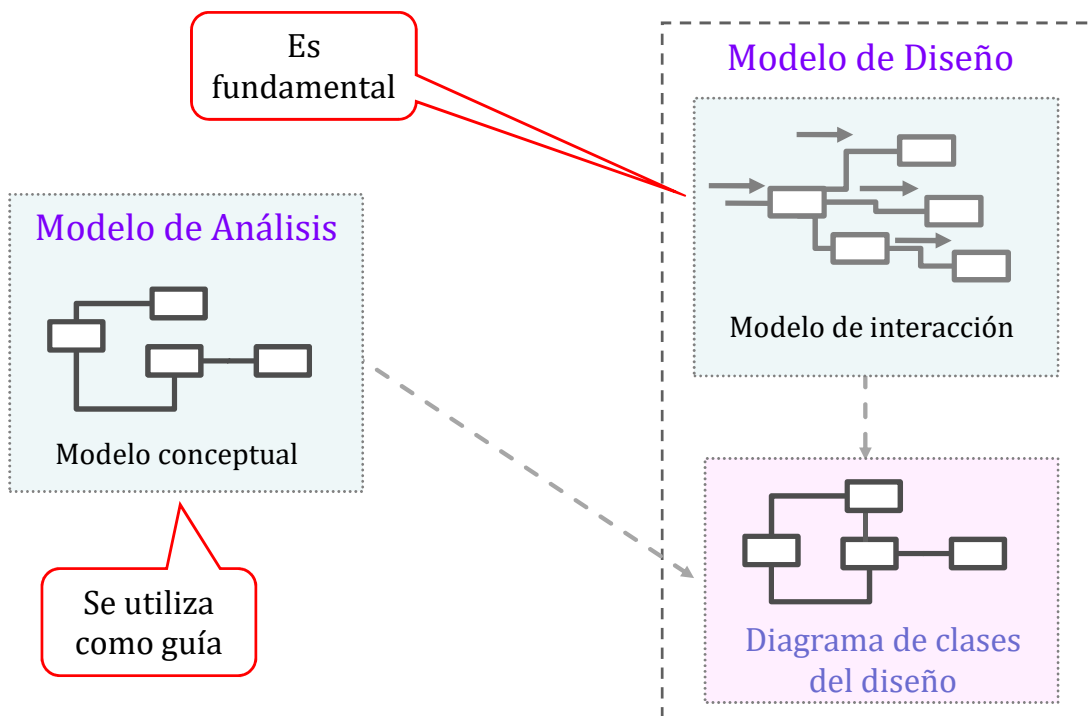
Contenido

Modelo de estructura de objetos

Elaboración del diagrama de clases del diseño

Modelo de estructura de objetos

¿De dónde partimos?



Modelo de estructura de objetos

Diagrama de clases del diseño

Describe gráficamente las especificaciones de la clases e interfaces software, y las relaciones entre estas en una aplicación. A diferencia del modelo conceptual representa la solución a un problema

Puede contener los siguientes elementos:

- Clases con sus atributos y operaciones
- Interfaces con sus operaciones y constantes
- Relaciones entre clases, entre interfaces o entre clases e interfaces
- Información sobre el tipo de los atributos y parámetros
- Navegabilidad de las asociaciones
- (cualquier elemento que forme parte de la solución)

Herramienta para su representación:

Diagrama de clases de UML

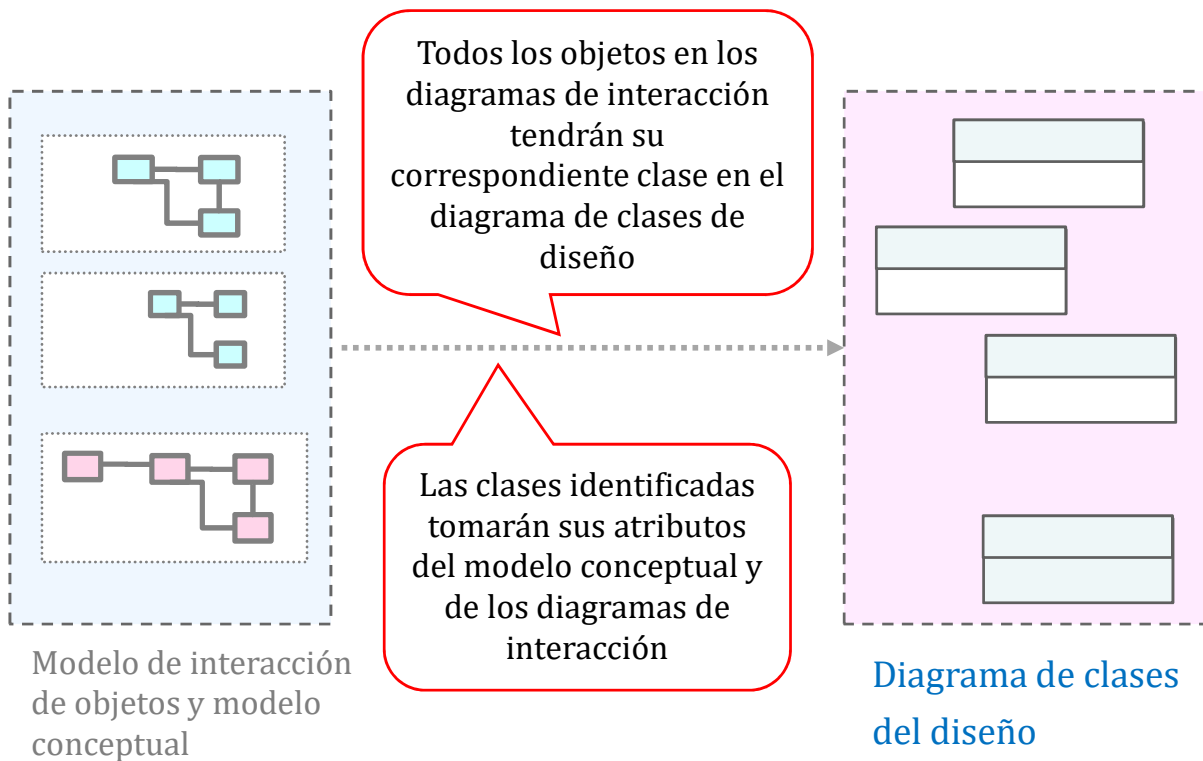
Elaboración del diagrama de clases del diseño

Pasos a seguir

1. Identificar y representar las clases
2. Identificar y añadir las operaciones
3. Añadir tipos de atributos y de parámetros
4. Identificar e incluir asociaciones y su navegabilidad
5. Identificar e incluir relaciones de dependencia
6. Incluir relaciones de generalización

Elaboración del diagrama de clases del diseño

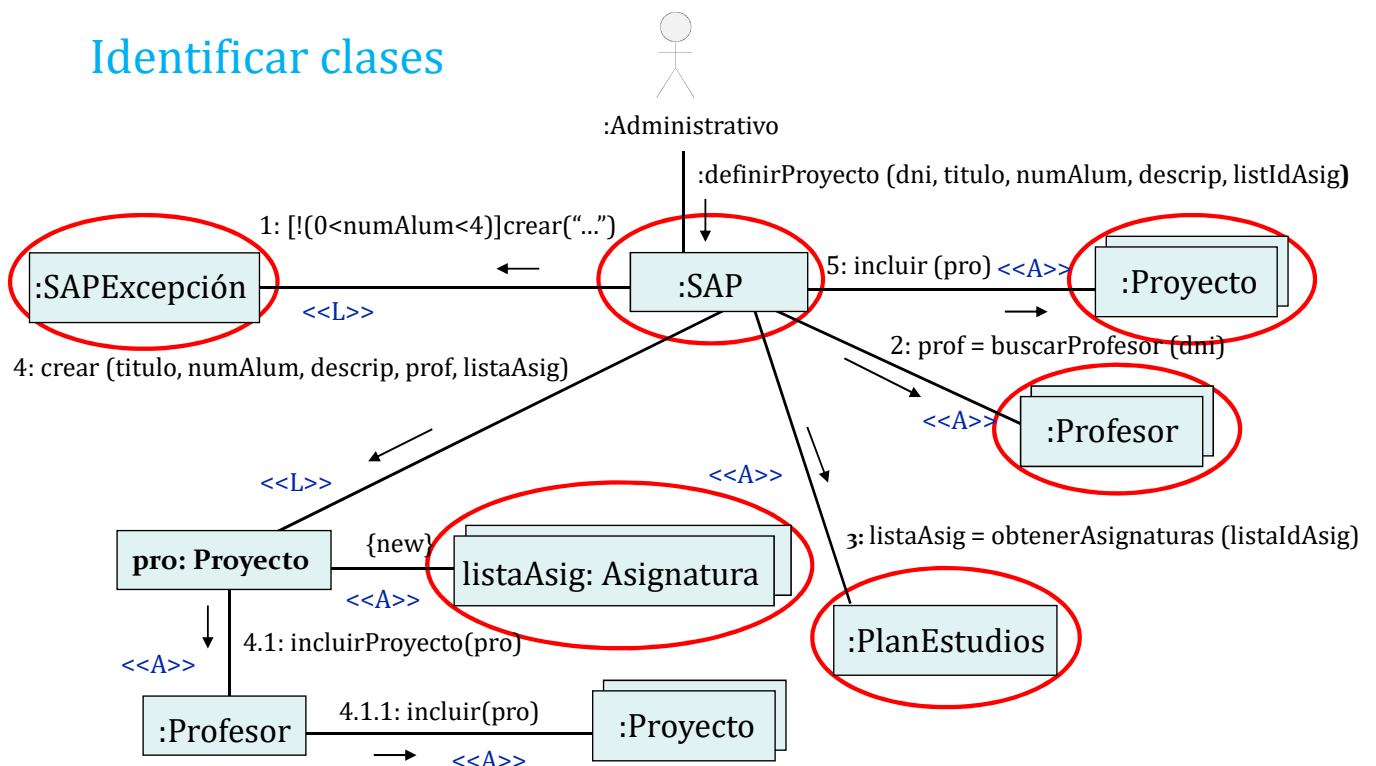
1. Identificar y representar las clases



Elaboración del diagrama de clases del diseño

1. Identificar y representar las clases

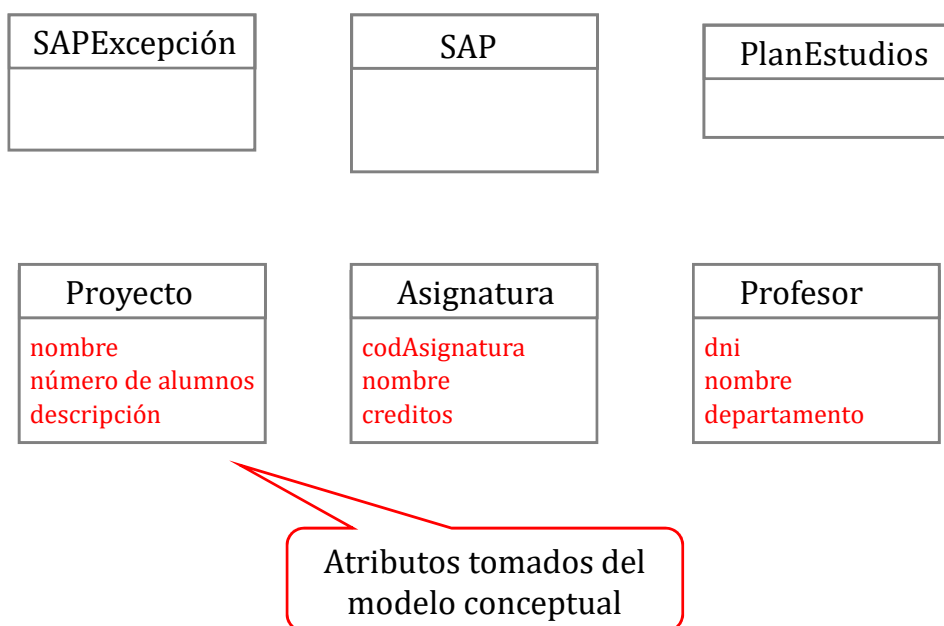
Identificar clases



Elaboración del diagrama de clases del diseño

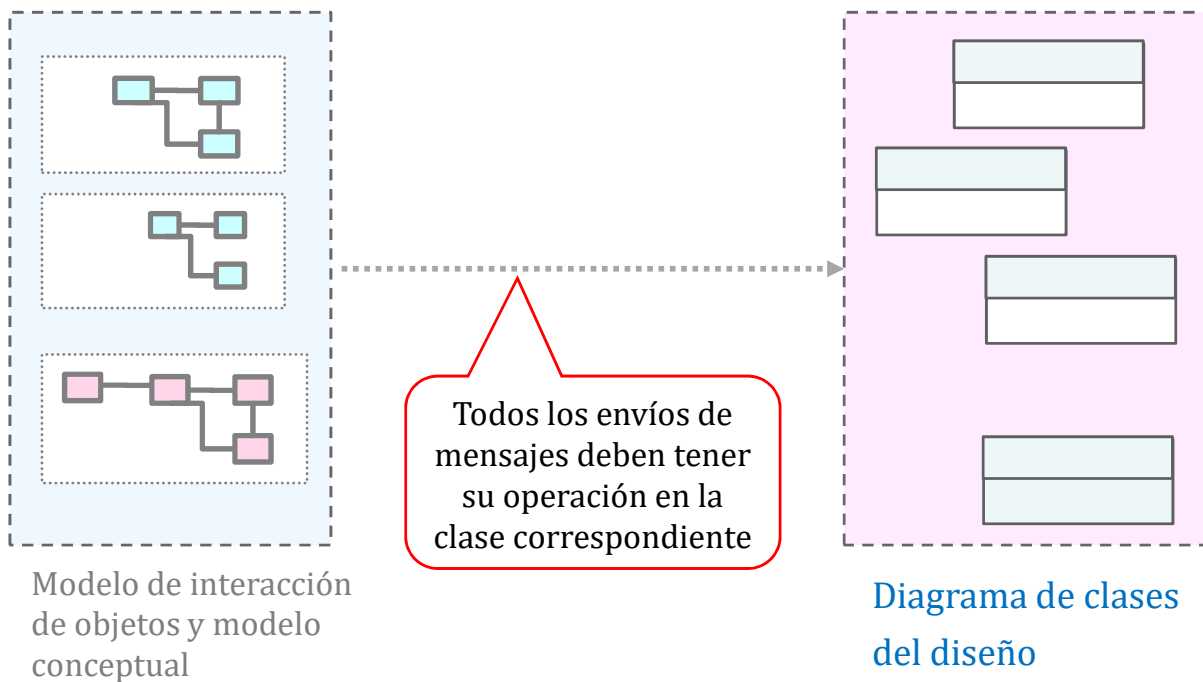
1. Identificar y representar las clases

Representar clases y sus atributos



Elaboración del diagrama de clases del diseño

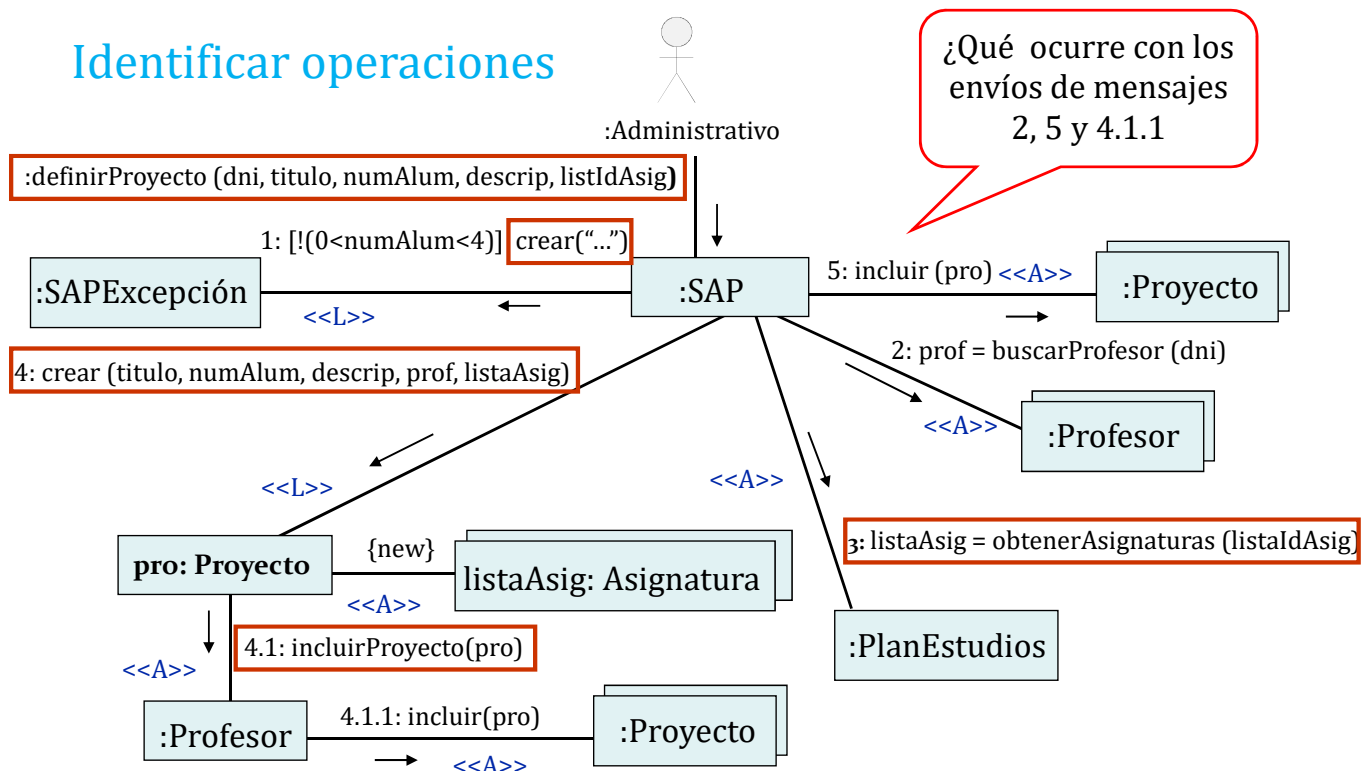
2. Identificar y añadir las operaciones



Elaboración del diagrama de clases del diseño

2. Identificar y añadir las operaciones

Identificar operaciones



Elaboración del diagrama de clases del diseño

2. Identificar y añadir las operaciones

Añadir operaciones

SAP
+ definirProyecto (dni, titulo, numAlum, descrip, lisIdAsig)

SAPException
+ crear (error)

PlanEstudios
~ obtenerAsignaturas (lisIdAsig)

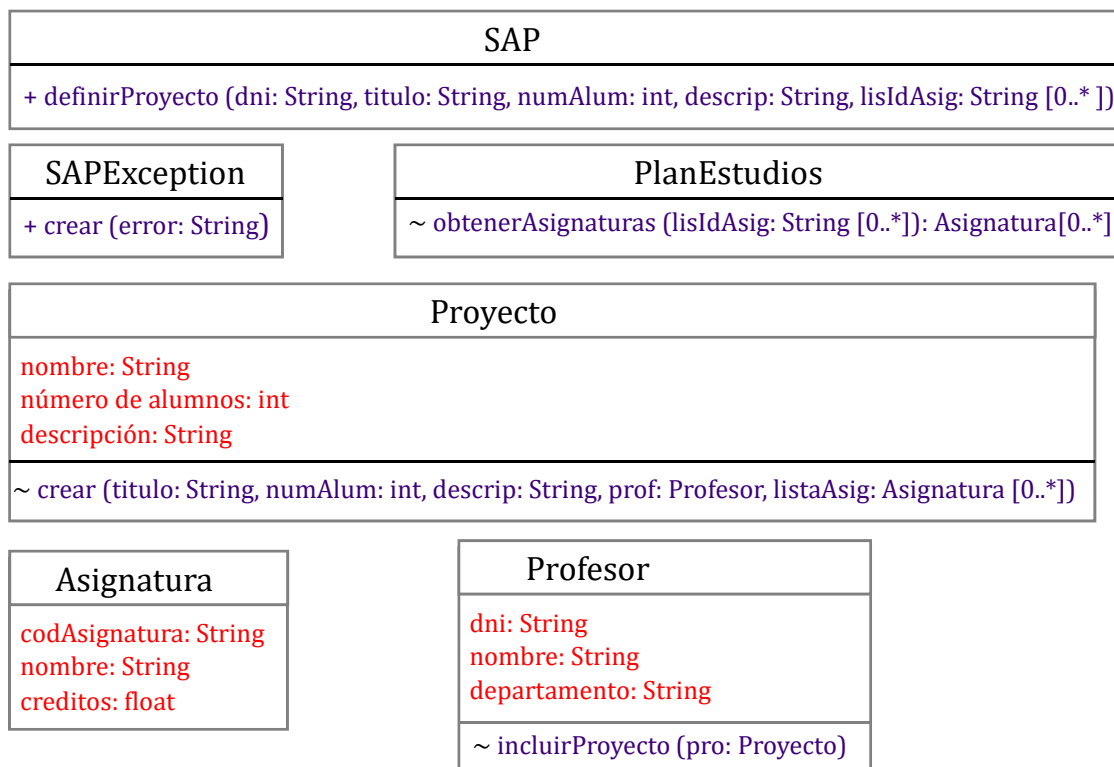
Proyecto
nombre número de alumnos descripción
~ crear (titulo, numAlum, descrip, prof, listaAsig)

Asignatura
codAsignatura nombre creditos

Profesor
dni nombre departamento
~ incluirProyecto (pro)

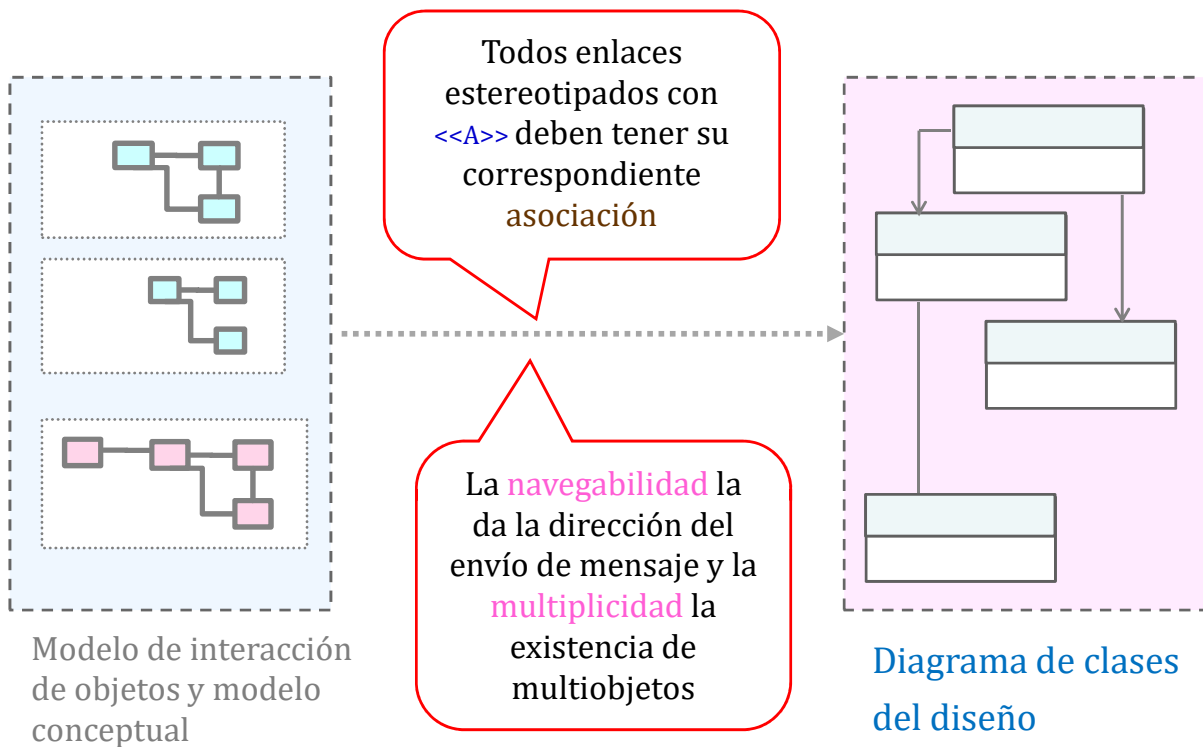
Elaboración del diagrama de clases del diseño

3. Añadir tipos de atributos y de parámetros



Elaboración del diagrama de clases del diseño

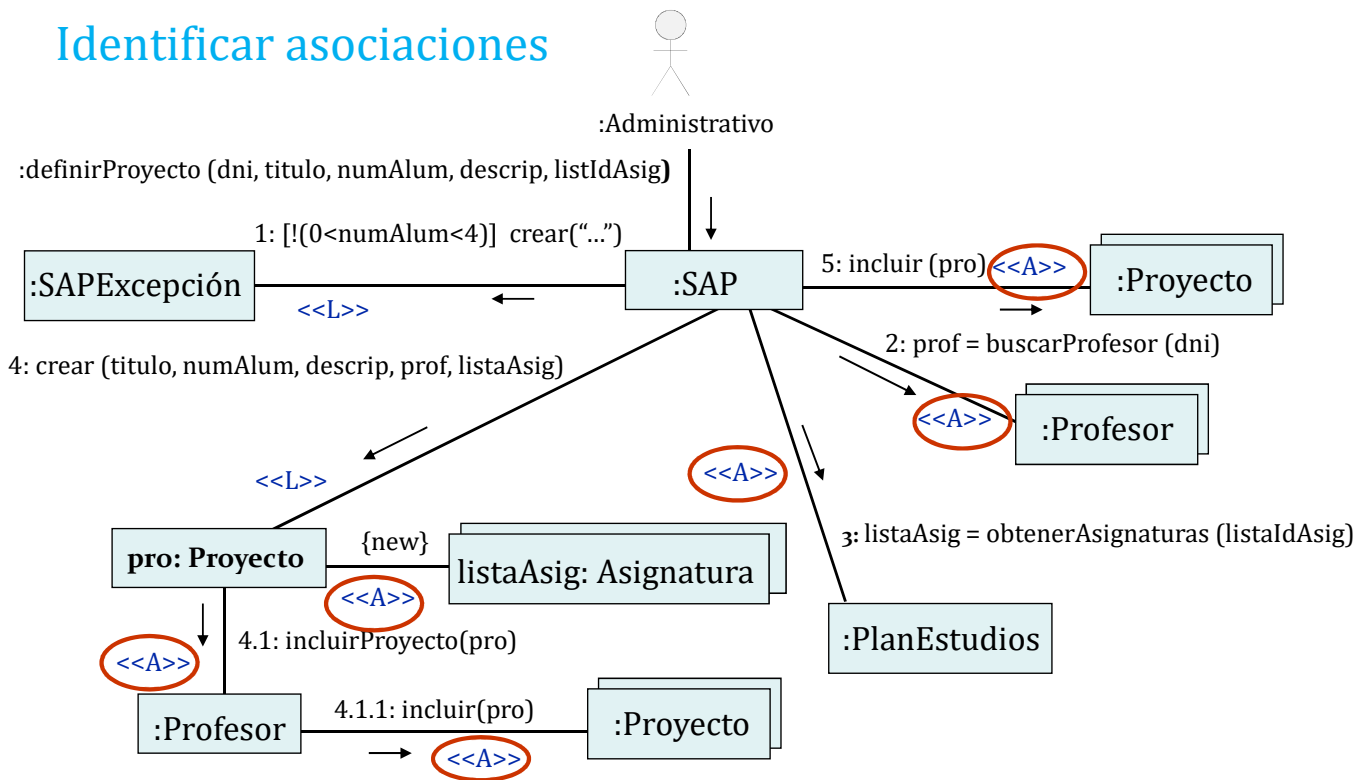
4. Identificar e incluir asociaciones y su navegabilidad



Elaboración del diagrama de clases del diseño

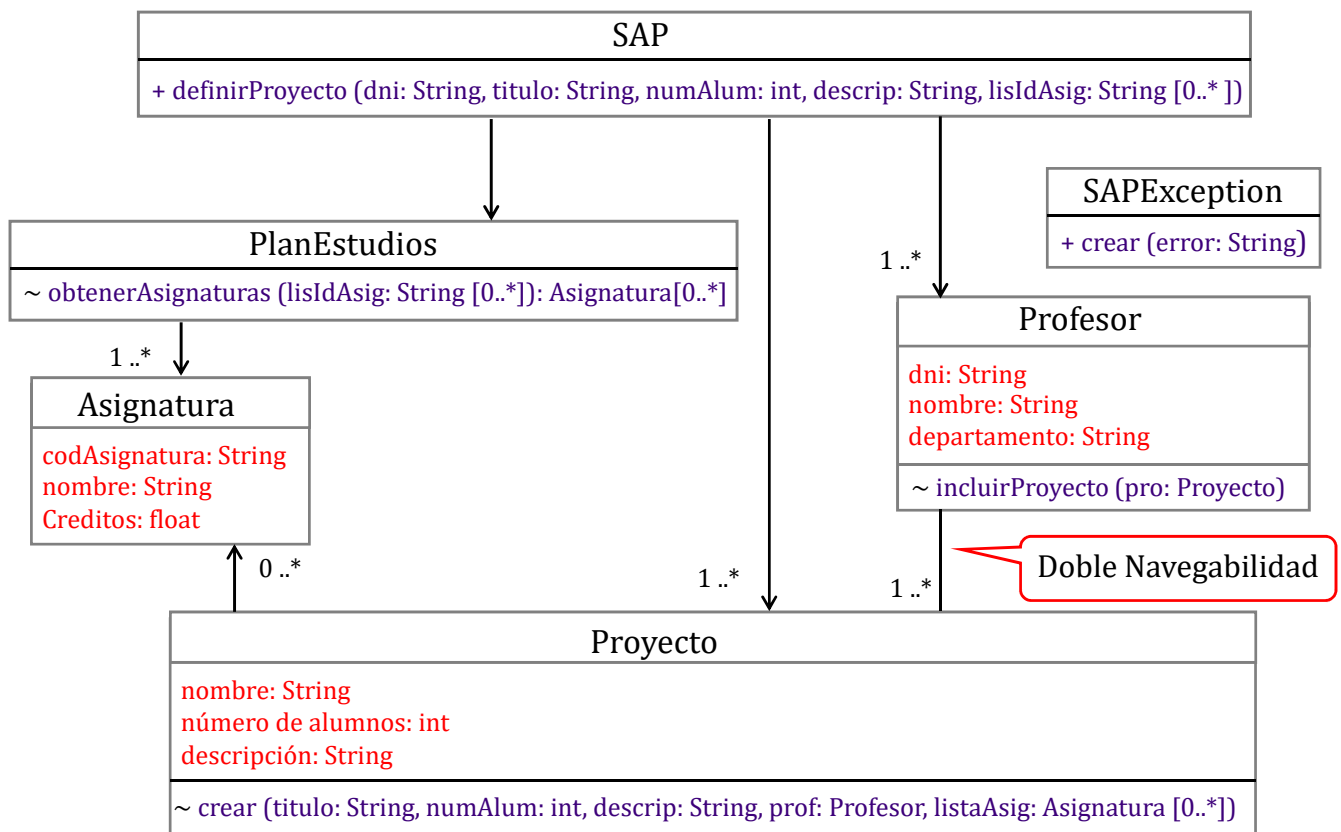
4. Identificar e incluir asociaciones y su navegabilidad

Identificar asociaciones



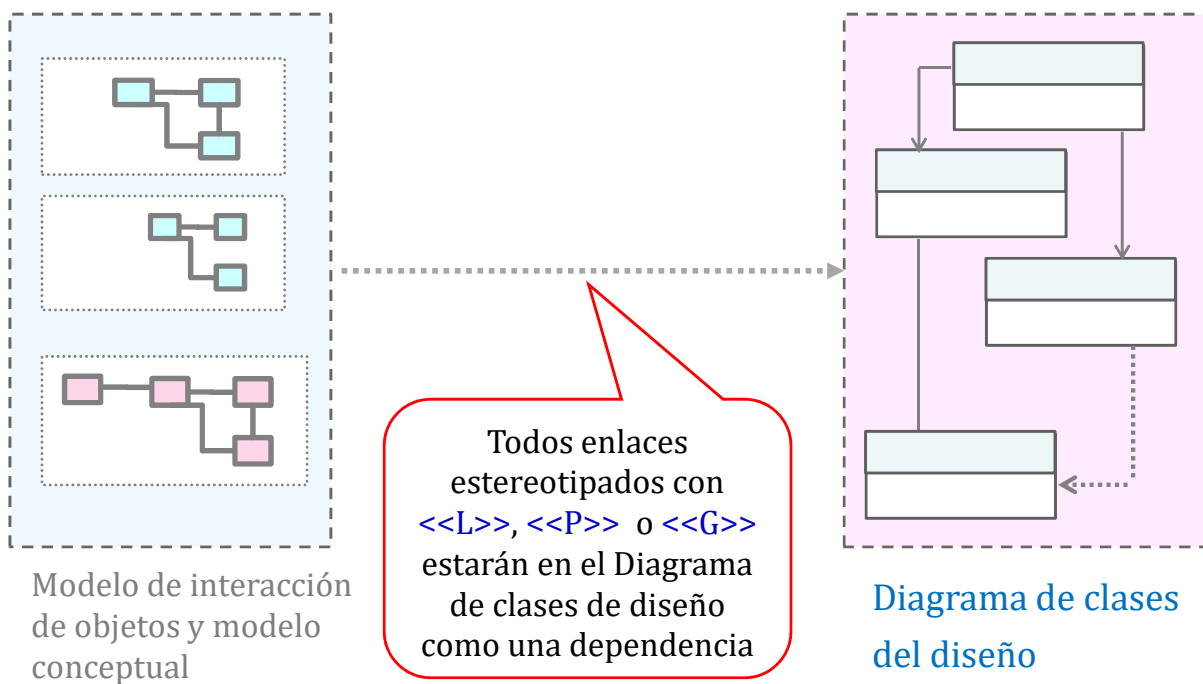
Elaboración del diagrama de clases del diseño

4. Identificar e incluir asociaciones y su navegabilidad



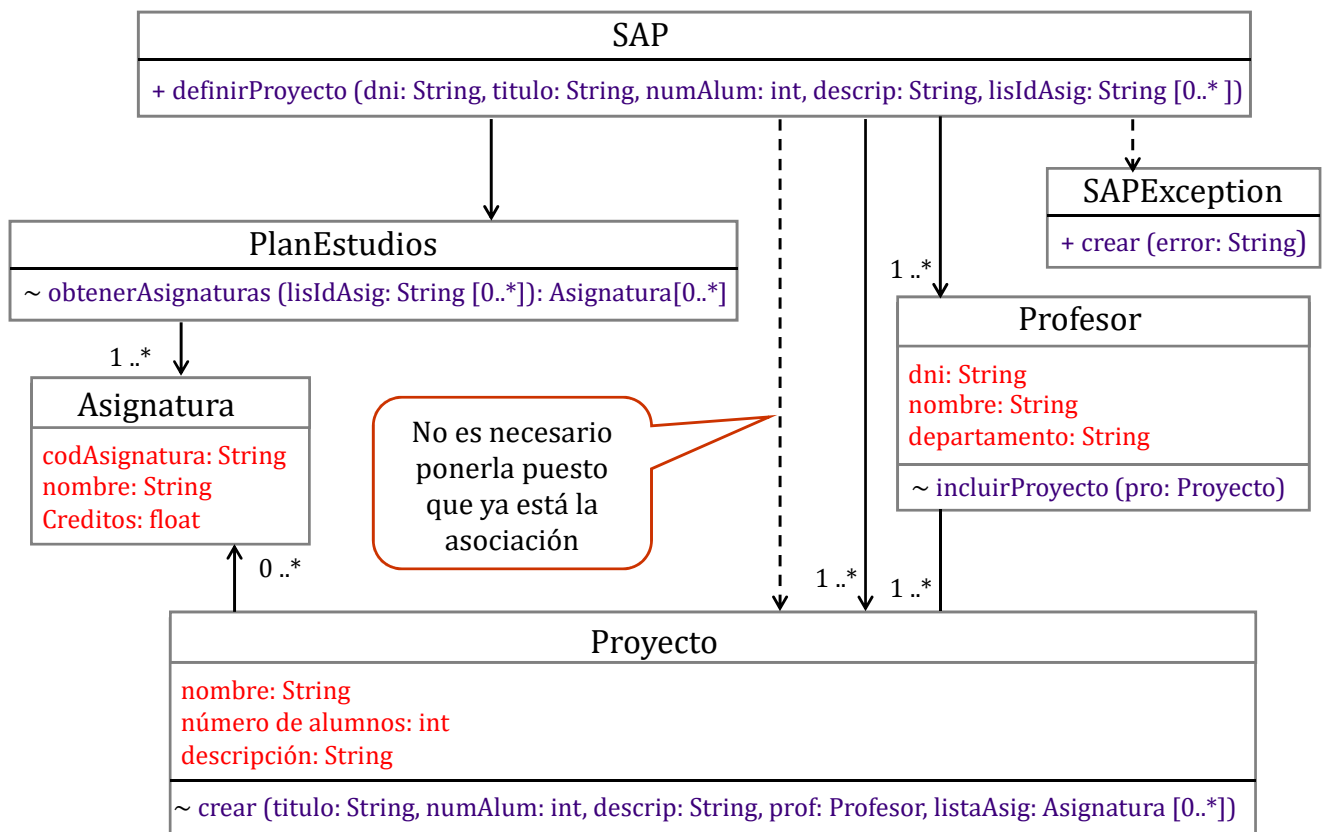
Elaboración del diagrama de clases del diseño

5. Identificar e incluir relaciones de dependencia



Elaboración del diagrama de clases del diseño

5. Identificar e incluir relaciones de dependencia



Elaboración del diagrama de clases del diseño

6. Incluir relaciones de generalización

Las generalizaciones que hay en el modelo conceptual también pueden aparecer en el diagrama de clases del diseño

Proceder de la siguiente forma:

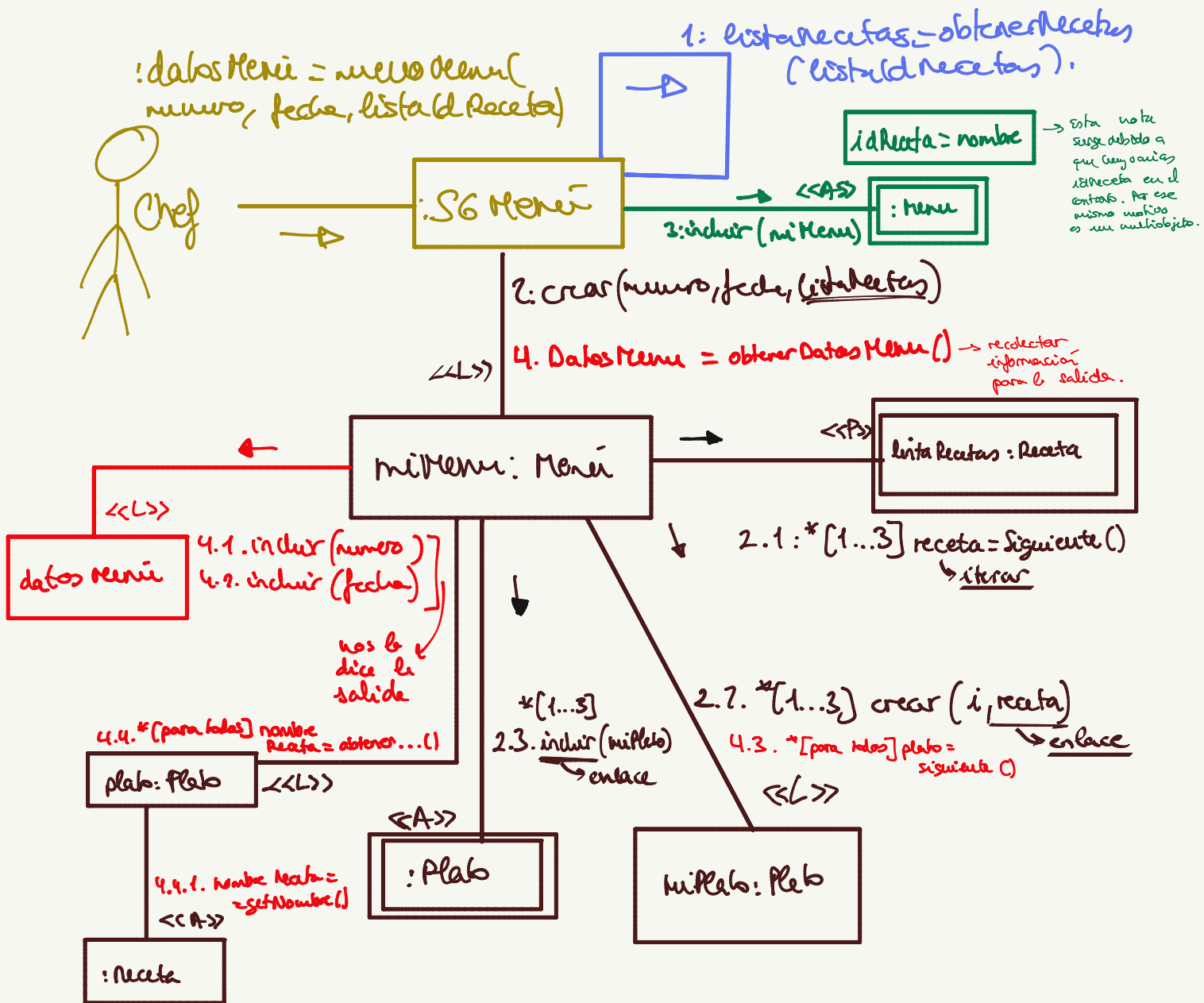
En el diagrama de clases del diseño obtenido, observar:

- Clases con nombres que identifiquen las distintas clasificaciones de un conjunto de objetos
- Clases con los mismos atributos
- Clases con la misma asociación con una clase
- Clases con operaciones con el mismo nombre o parecido

Para asegurar que se corresponde con igual o parecida semántica, mirar la similitud de estructura de los diagramas de comunicación correspondientes

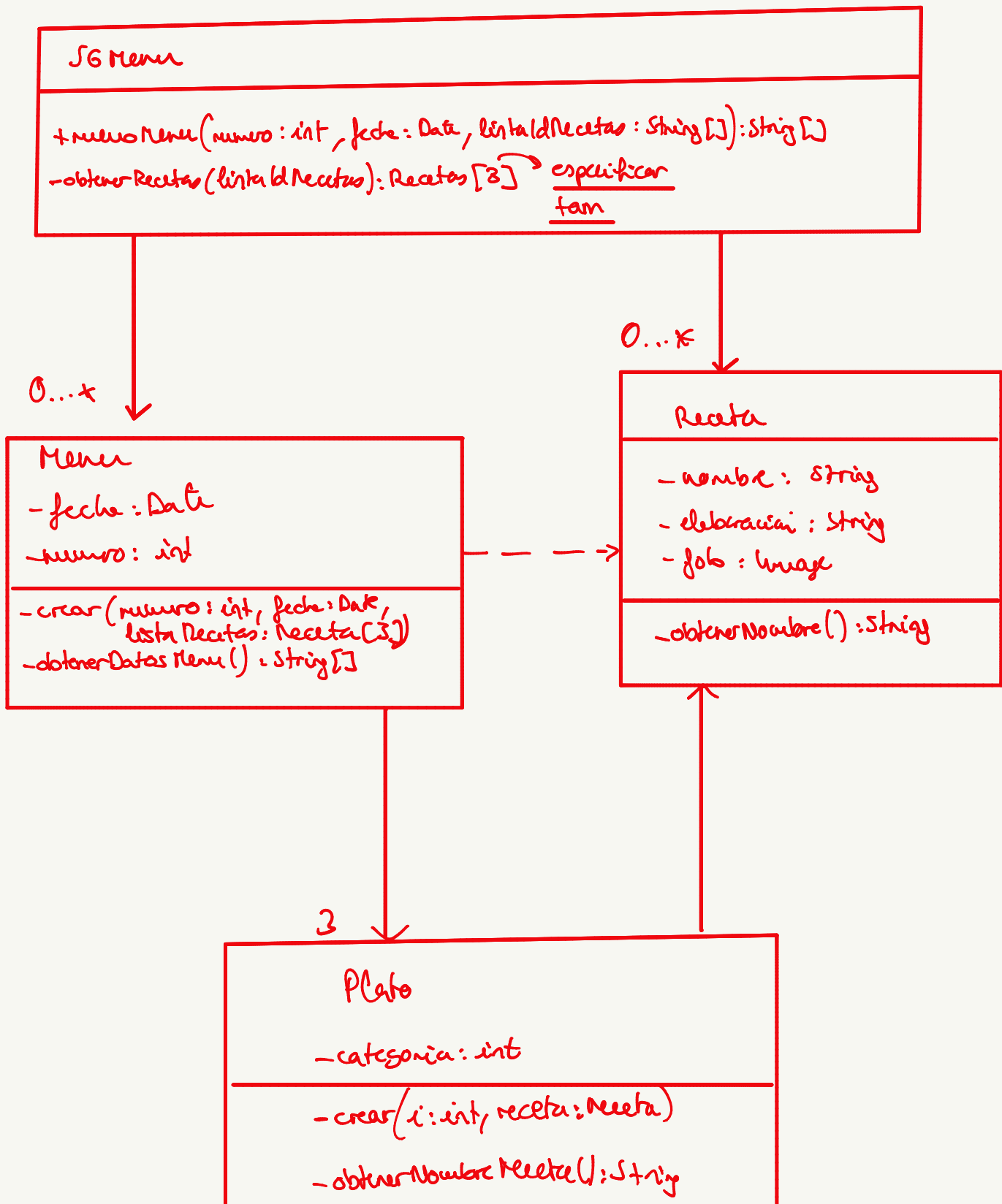
Si se da una o varias de estas situaciones establecer una generalización entre las clases, llevando a la superclase atributos, operaciones y asociaciones comunes

DIAGRAMA DE COMUNICACIÓN



Recorremos info para la salida.

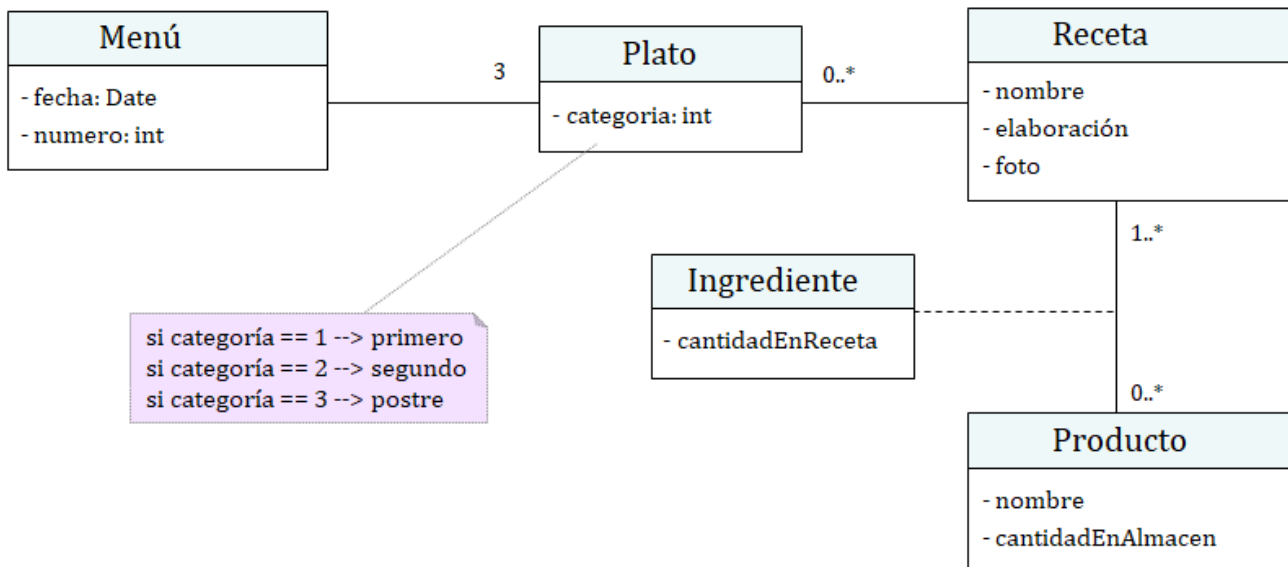
ORDENADA DE CLASES



Tomando como base el modelo conceptual y el contrato para la operación nuevoMenú que se adjuntan:

- Realice el diagrama de comunicación de la operación nuevoMenú.
- Partiendo del diagrama anterior, realice el diagrama de clases de diseño

Modelo conceptual:



Contrato:

Nombre	nuevoMenú (número, fecha, listaIdReceta)
Responsabilidad	Incluir un nuevo menú, junto con las recetas que le corresponde a cada plato: primero, segundo y postre.
Tipo	SGMenú (Sistema)
Notas	listaIdReceta [1] == idReceta para el primero listaIdReceta [2] == idReceta para el segundo listaIdReceta [3] == idReceta para el postre
Referencias	
Excepciones	- No existe alguno de los objetos de la clase <i>Receta</i> identificado por los elementos de <i>listaIdReceta</i> .
Salida	- La <i>fecha</i> y el <i>número</i> del <i>Menú</i> Para todos los <i>Platos</i> - El <i>nombre</i> de la <i>Receta</i> correspondiente
Precondiciones	
Poscondiciones	<ul style="list-style-type: none"> Se creó un objeto de la clase <i>Menú</i>, <i>miMenú</i>, debidamente inicializado. De 1 a 3 <ul style="list-style-type: none"> Se creó un objeto de la clase <i>Plato</i>, <i>miPlato</i>, debidamente inicializado. Se creó un enlace entre <i>miPlato</i> y el objeto <i>Receta</i> identificado por el correspondiente elemento de <i>listaIdReceta</i>. Se creó un enlace entre el objeto <i>miPlato</i> y el objeto <i>miMenú</i>.

Diagrama de comunicación:

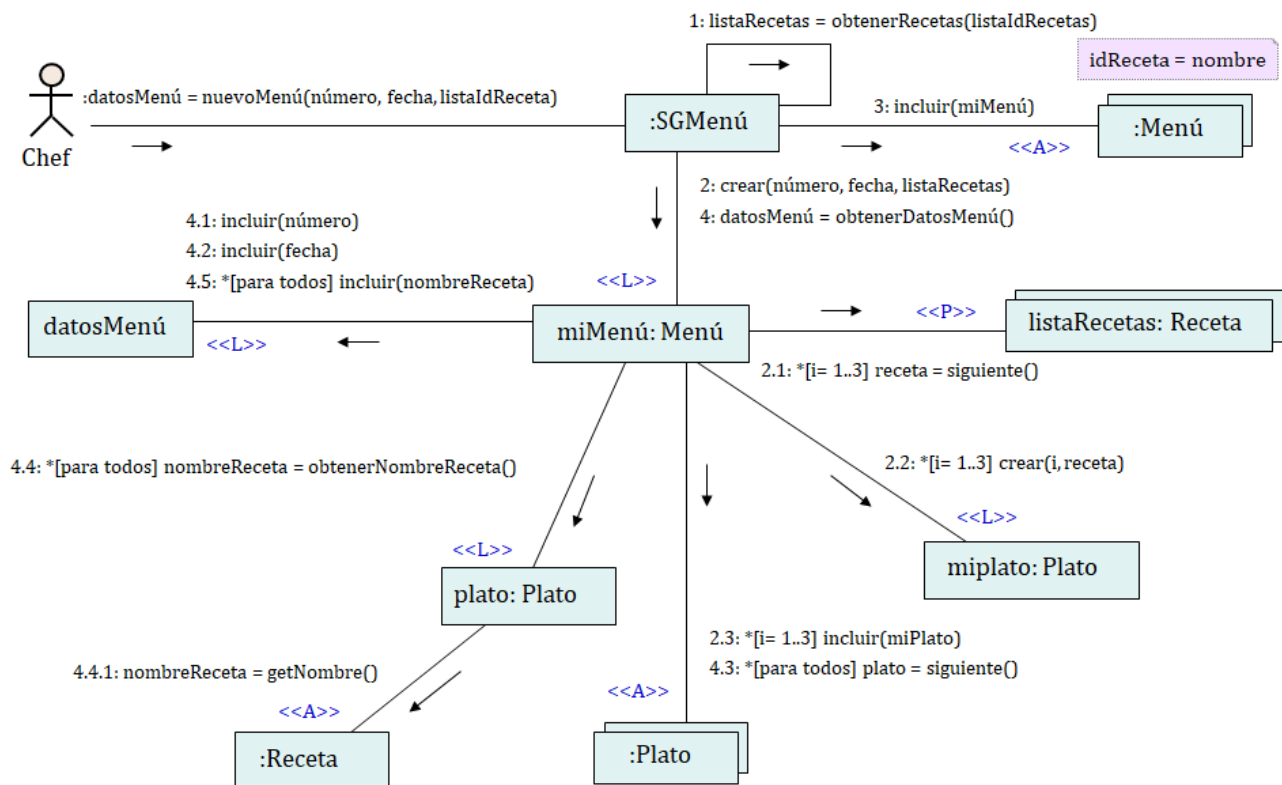
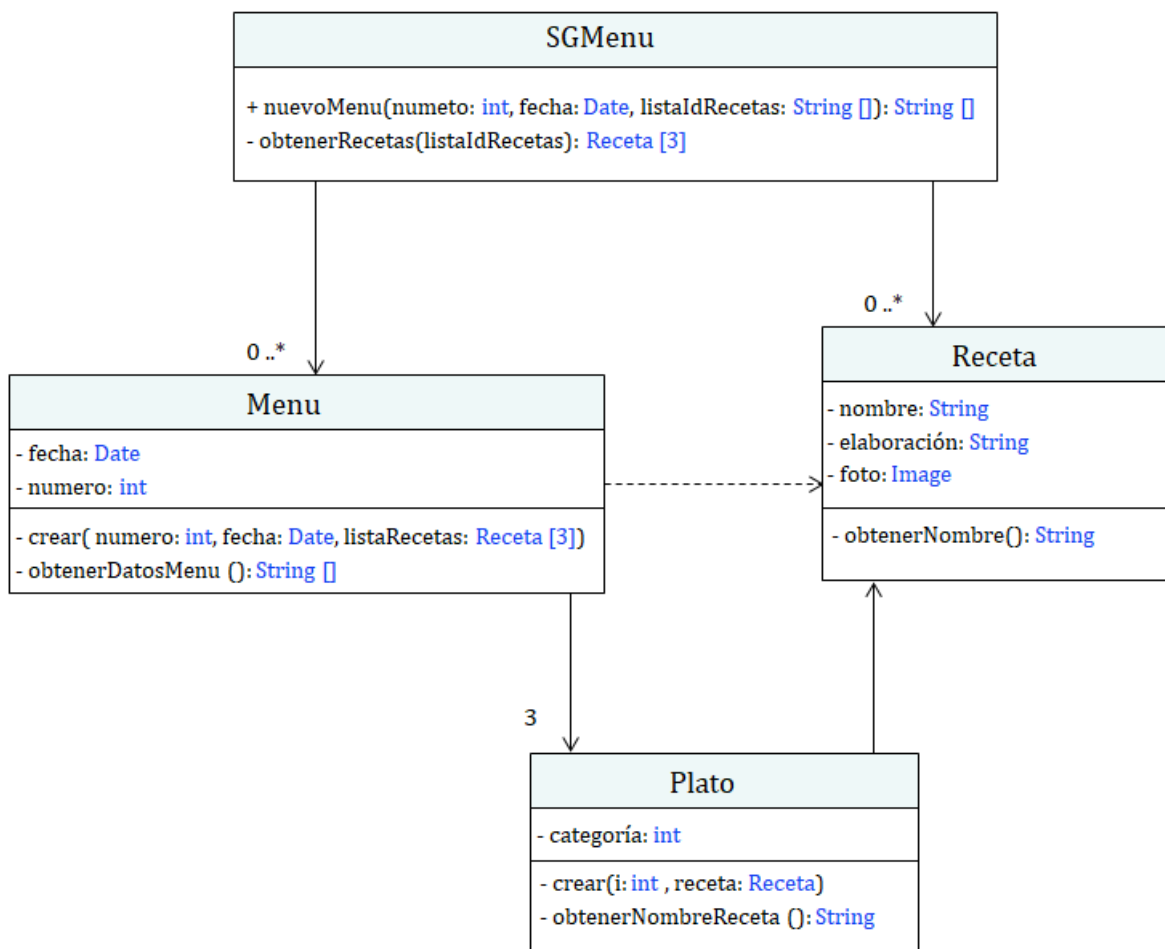


Diagrama de clases del diseño:



GESTIÓN DE UN VIDEOCLUB

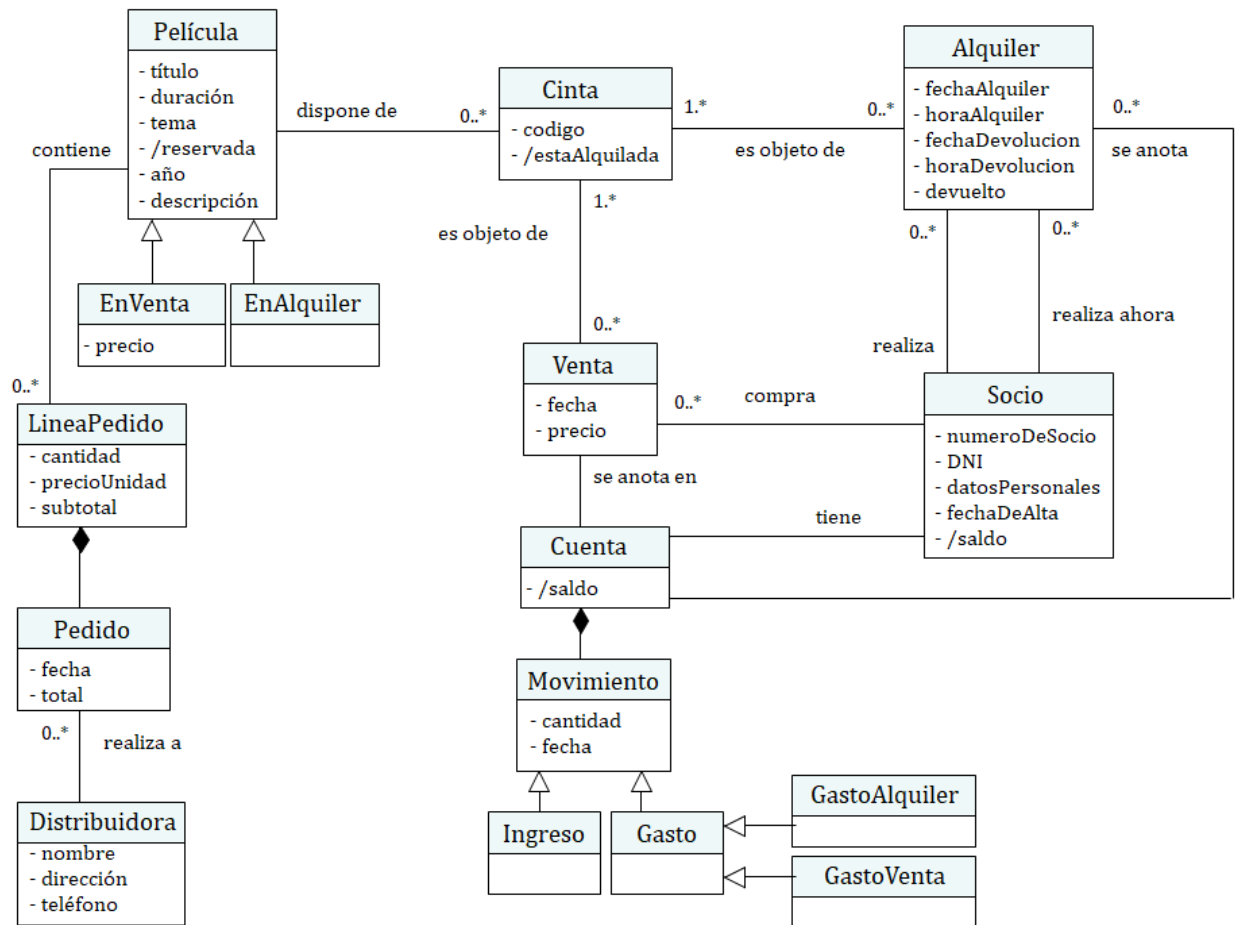
MODELO DEL DISEÑO

Versión 2.0

Índice

1. Diagrama de conceptos
2. Diagramas de comunicación
3. Diagrama de clases del diseño

1. Diagrama de conceptos



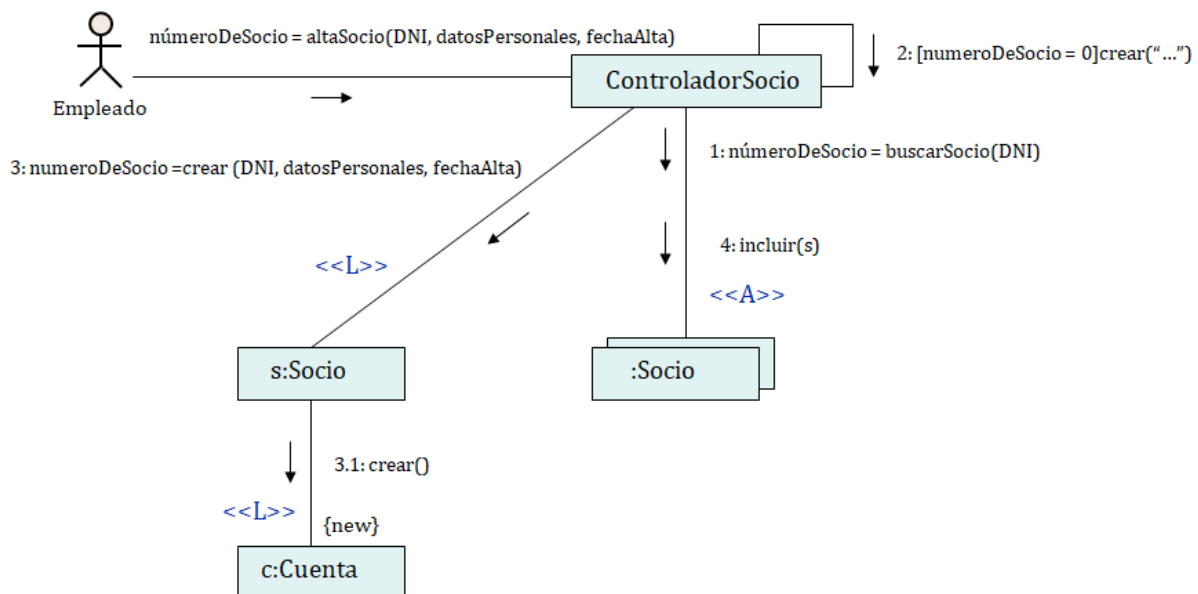
2. Diagramas de comunicación

Alta socio

Contrato

Nombre	numeroDeSocio = altaSocio(DNI, datosPersonales, fechaAlta)
Responsabilidad	Dar de alta un cliente como socio del videoclub.
Tipo	Videoclub (sistema)
Referencias Cruzadas	Caso de uso Alta de Socio (CU-6)
Notas	
Excepciones	- El cliente ya es socio del videoclub.
Salida	numeroDeSocio
Precondiciones	
Poscondiciones	<ul style="list-style-type: none"> - Se creó un objeto de la clase Socio, soc, debidamente inicializado, con fechaAlta la fecha actual. - Se creó un objeto de la clase Cuenta, cue, con saldo a 0. - Se creó un enlace entre soc y cue.

Diagrama de comunicación

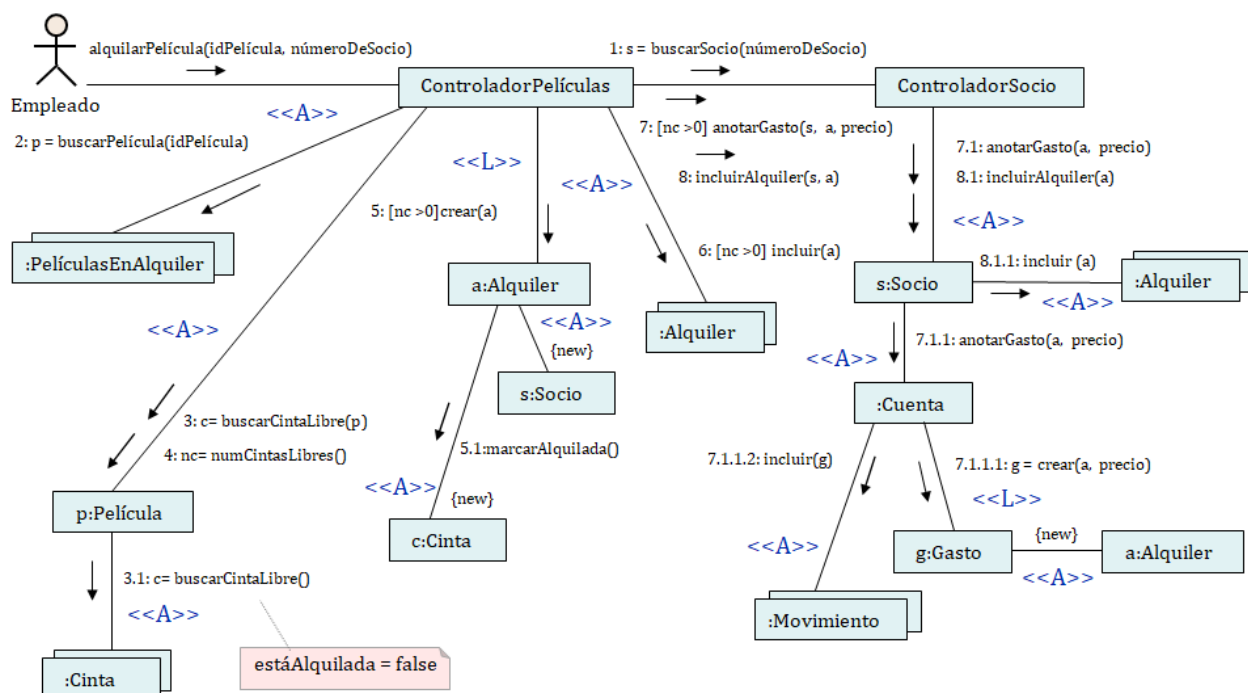


Alquilar Película

Contrato

Nombre	alquilarPelícula(idPelícula, numeroDeSocio)
Responsabilidad	Realizar un alquiler de una película por parte de un socio.
Tipo	Videoclub (sistema)
Referencias cruzadas	Caso de uso Alquilar Película (CU-18)
Notas	
Excepciones	<ul style="list-style-type: none"> - No existe el Socio con número de socio numeroDeSocio. - No existen copias disponibles de la Película identificada por idPelícula.
Salida	
Precondiciones	
Poscondiciones	<ul style="list-style-type: none"> - Se creó un objeto de la clase Alquiler, alq, con fechaAlquiler la fecha actual y horaAlquiler la hora actual. - Se creó un enlace “realiza ahora” entre alq y el objeto de la clase Socio identificado por numeroDeSocio. - Se creó un enlace “realiza” entre alq y el objeto de la clase Socio identificado por numeroDeSocio. - Se creó un enlace entre alq y el objeto de la clase Cinta correspondiente a la Película identificada por idPelícula. - Se modifico el atributo estáAlquilada del objeto Cinta al valor “TRUE”. - Se creó un objeto de la clase GastoAlquiler, ga, con fecha la fecha actual y cantidad el precio de un día de alquiler. - Se creó un enlace entre ga y alq.

Diagrama de comunicación

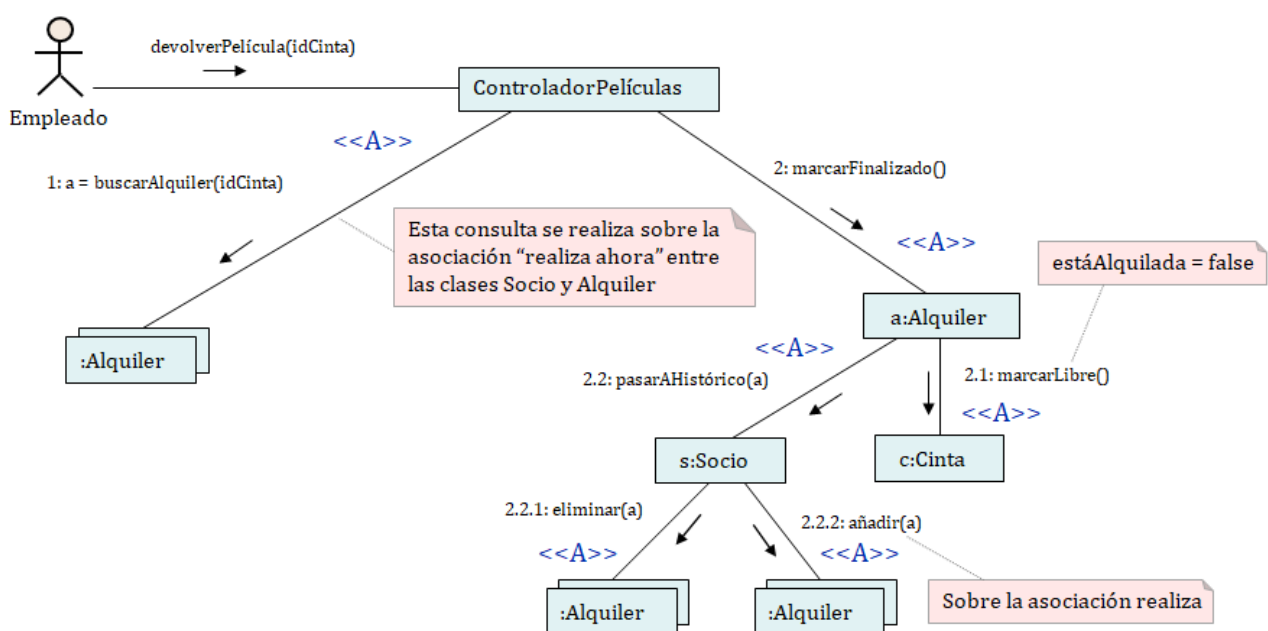


Devolver Película

Contrato

Nombre	devolverPelícula(idCinta)
Responsabilidad	Devolver una película que se encuentra alquilada.
Tipo	Videoclub(sistema)
Referencias cruzadas	Caso de uso Devolver Película (CU-20)
Notas	
Excepciones	<ul style="list-style-type: none">- Los datos no se corresponden con un socio del videoclub.- El tiempo de alquiler sobrepasa el pago realizado (se aplica nueva tarifa).
Salida	
Precondiciones	
Poscondiciones	<ul style="list-style-type: none">- Se modificó el atributo fechaDevolucion, correspondiente al objeto Alquiler asociado al objeto de la clase Cinta identificado por idCinta, al valor de la fecha actual.- Se modificó el atributo horaDevolucion, correspondiente al objeto Alquiler asociado al objeto de la clase Cinta, identificado por idCinta, al valor de la hora actual.- Se modificó el atributo devuelto correspondiente al objeto Alquiler asociado al objeto de la clase Cinta, identificado por idCinta, al valor "TRUE".- Se modificó el atributo estaAlquilada del objeto de la clase Cinta identificado por idCinta, al valor "FALSE".- Se eliminó el enlace "realiza ahora" entre el Socio que ha devuelto la cinta y el objeto de la clase Alquiler correspondiente.

Diagrama de comunicación

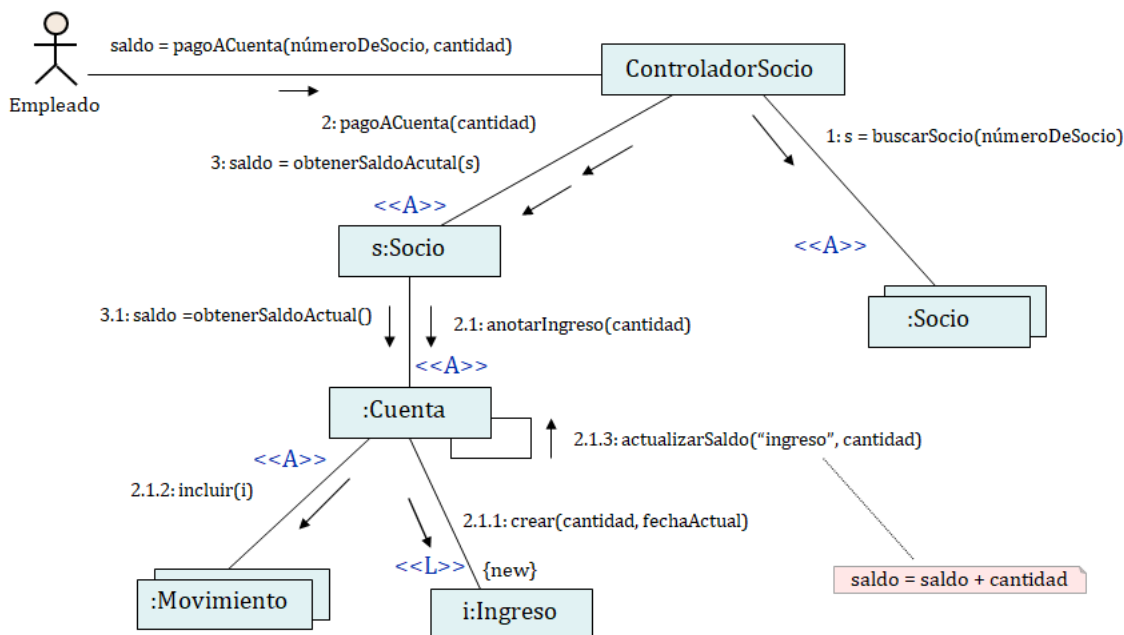


Pago a Cuenta

Contrato

Nombre	saldo= pagoACuenta(numeroDeSocio, cantidad)
Responsabilidad	Realizar un ingreso en la cuenta de un socio
Tipo	Videoclub (sistema)
Referencias cruzadas	Caso de uso Pago a Cuenta (CU-13)
Notas	
Excepciones	
Salida	Saldo actualizado en la cuenta del socio.
Precondiciones	
Poscondiciones	<ul style="list-style-type: none"> - Se creó un objeto de la clase Ingreso, ing, con la fecha actual y la cantidad indicada en el argumento de entrada. - Se creó un enlace entre ing y el objeto de la clase Cuenta asociada al objeto Socio identificado por NumeroDeSocio. - Se modificó el atributo saldo del objeto Cuenta asociada al objeto Socio identificado por NumeroDeSocio al valor saldo + cantidad.

Diagrama de comunicación

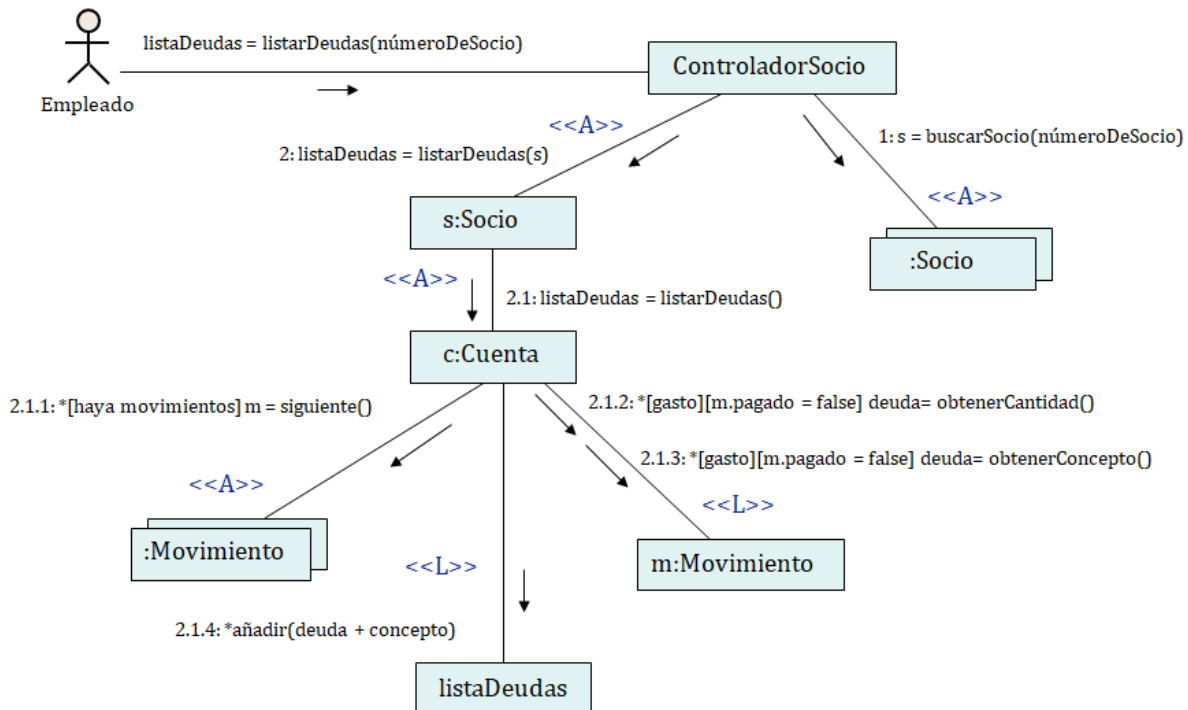


Listar Deudas

Contrato

Nombre	listaDeudas=listarDeudas(numeroDeSocio)
Responsabilidad	Obtener una lista de las deudas de un Socio
Tipo	Videoclub (sistema)
Notas	
Excepciones	- No existe el Socio con número de socio numeroDeSocio
Salida	Lista de {cantidad, concepto} para todos los movimientos de tipo Pago que no haya abonado el Socio con número de Socio numeroDeSocio
Precondiciones	
Poscondiciones	

Diagrama de comunicación



Observaciones respecto a los diagramas de comunicación

1. Por cuestiones de legibilidad, en los diagramas de comunicación, se han considerado, como controladores, clases diferentes a la clase que representa al sistema
2. La forma de representar una estructura condicional en un diagrama de comunicación es :
`[condición] operación;` que significa que si se verifica la `condición` se llevará a cabo la `operación`. Por ejemplo, `[nc > 0] incluir(a)`.
3. La forma de representar una estructura repetitiva en un diagrama de comunicación es:
`*[condición] operación;` que significa que mientras se verifique la `condición` se llevará a cabo la operación . Por ejemplo, `*[m.pagado= false] deuda=obtenercantidad()`
4. La forma de recorrer los elementos de una lista en un diagrama de comunicación es:
`*[condición] n= siguiente()`, que significa que mientras se cumpla la `condición` se pasa al elemento siguiente de la lista. Por ejemplo, `*[haya movimientos] n=siguiente()`
5. Cuando la salida de una operación sea una lista de elementos, esa lista se puede considerar como un objeto en el diagrama de comunicación (ver diagrama de la operación `listarDeudas`)
6. Al igual que se pueden utilizar en un diagrama de comunicación variables locales, también se puede usar, si es necesario, objetos locales, es decir objetos (que pueden ser o no de una clase particular) que pueden servir, por ejemplo, para almacenar localmente datos complejos.

3. Diagrama de clases del diseño

