



ANÁLISIS Y DISEÑO 0.0

Prof. Jannelly Bello



LENGUAJE UNIFICADO DE MODELADO UML

Prof. Jannelly Bello

¿QUÉ ES UML?

- Lenguaje Unificado de Modelado
- Permite Visualizar, Especificar, Construir, y Documentar software orientado a objetos.
- Divide el sistema en un conjunto de diagramas que a su vez representan una vista del proyecto.
- Todos los diagramas muestran en conjunto la arquitectura del proyecto.



ORIGEN

- La notación UML se deriva y unifica las tres metodologías de análisis y diseño Orientado a objetos: **Grady Booch, James Rumbaugh e Ivar Jacobson.**
- El desarrollo de UML comenzó a finales de 1994 cuando Grady Booch y Jim Rumbaugh de Rational Software Corporation empezaron a unificar sus métodos. A finales de 1995, Ivar Jacobson y su compañía Objectory se incorporaron a Rational en su unificación, aportando el método ISOO (Ingeniería del Software Orientada a Objetos).
- Las metodologías de Booch y Rumbaugh se enfocan hacia el modelado de los objetos que conforman el sistema, su relación y colaboración.

ORIGEN

- La metodología de Jacobson se enfoca en el usuario. Todo en su método se deriva de los casos de uso.
- UML se ha ido fomentando y aceptando como estándar desde la formación de OMG (**Object Management Group - *Grupo de Gestión de Objetos***).
- En 1997 UML 1.1 fue aprobada por la OMG convirtiéndose en la notación estándar para el análisis y el diseño orientado a objetos.
- En la actualidad ya se publicó la revisión 2.0.



PRINCIPIOS

- ❑ La forma como vemos el problema tiene una profunda influencia en la forma como acometemos el problema y le damos solución al mismo.
- ❑ Para modelar un sistema complejo no es suficiente un único modelo se requieren múltiples modelos donde cada uno representa una vista del sistema; estos modelos se complementan entre si.
- ❑ Cualquier modelo puede ser representado con diferentes grados de precisión.
- ❑ Los mejores Modelos están ligados a la realidad.



DIAGRAMAS

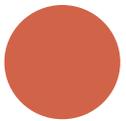
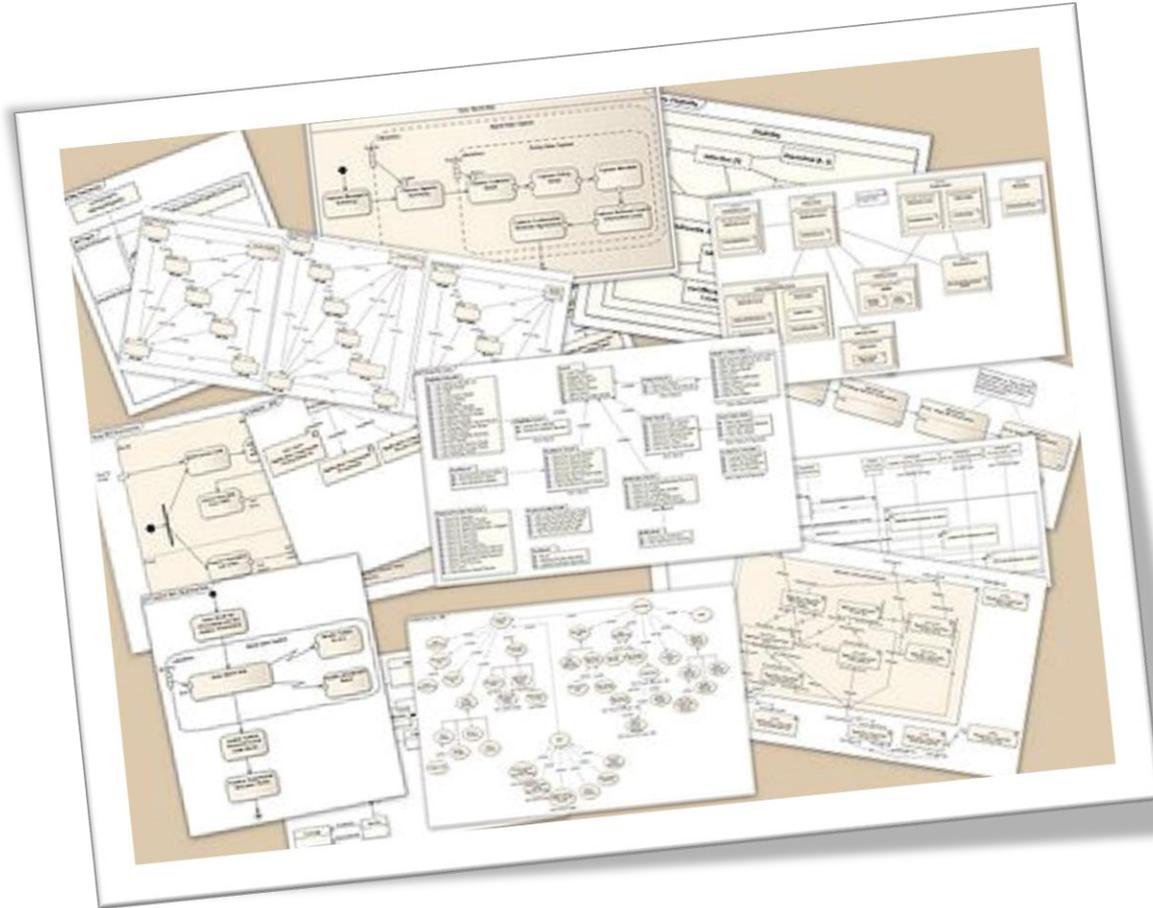


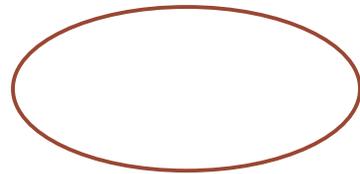
DIAGRAMA DE CASOS DE USO

¿Qué es un Caso de Uso?

- Describe la funcionalidad del sistema desde el punto de vista del usuario, es decir, se refiere al uso que se le dará al sistema.

Notación:

Son representados por un óvalo y nombrados siempre con verbos en infinitivo. Ejemplo: Inscribir a Curso.



Inscribir a Curso



DIAGRAMA DE CASOS DE USO

¿Qué es un Actor?

- Agente externo que interactúa con el sistema. Representa un conjunto coherente de roles que juegan los usuarios de los casos de uso al interactuar con el sistema.

Notación:

Representados por una figura de alambre de una persona.

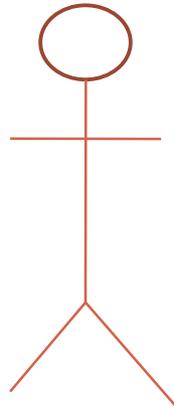


DIAGRAMA DE CASOS DE USO

¿Qué es un Diagrama de Caso de Uso?

- ❑ Describe las funcionalidades del sistema a partir de las interacciones del usuario.

Relaciones entre Casos de Uso:

- ❑ **Asociación:** Comunicación entre un actor y un caso de uso donde participa. Notación: 
- ❑ **Inclusión:** Relación de dependencia entre dos casos de uso que denota la inclusión del comportamiento de un escenario en otro. Se utiliza cuando existen casos de uso cuyo comportamiento es común a varios casos de uso. Se representa incluyendo una etiqueta <<include>> sobre la flecha que conduce al caso de uso que se incluye.



DIAGRAMA DE CASOS DE USO

Extensión: Relación de dependencia entre dos casos de uso que denota que un caso de uso es una especialización de otro, es decir, incluye comportamiento bajo determinadas condiciones.

Un caso de uso extiende a otro cuando sin alterar a este, se incorpora su funcionalidad como parte integral del primero. Se denota con una relación que apunta del caso extendido al caso base incluyendo una etiqueta <<extend>> sobre la flecha.



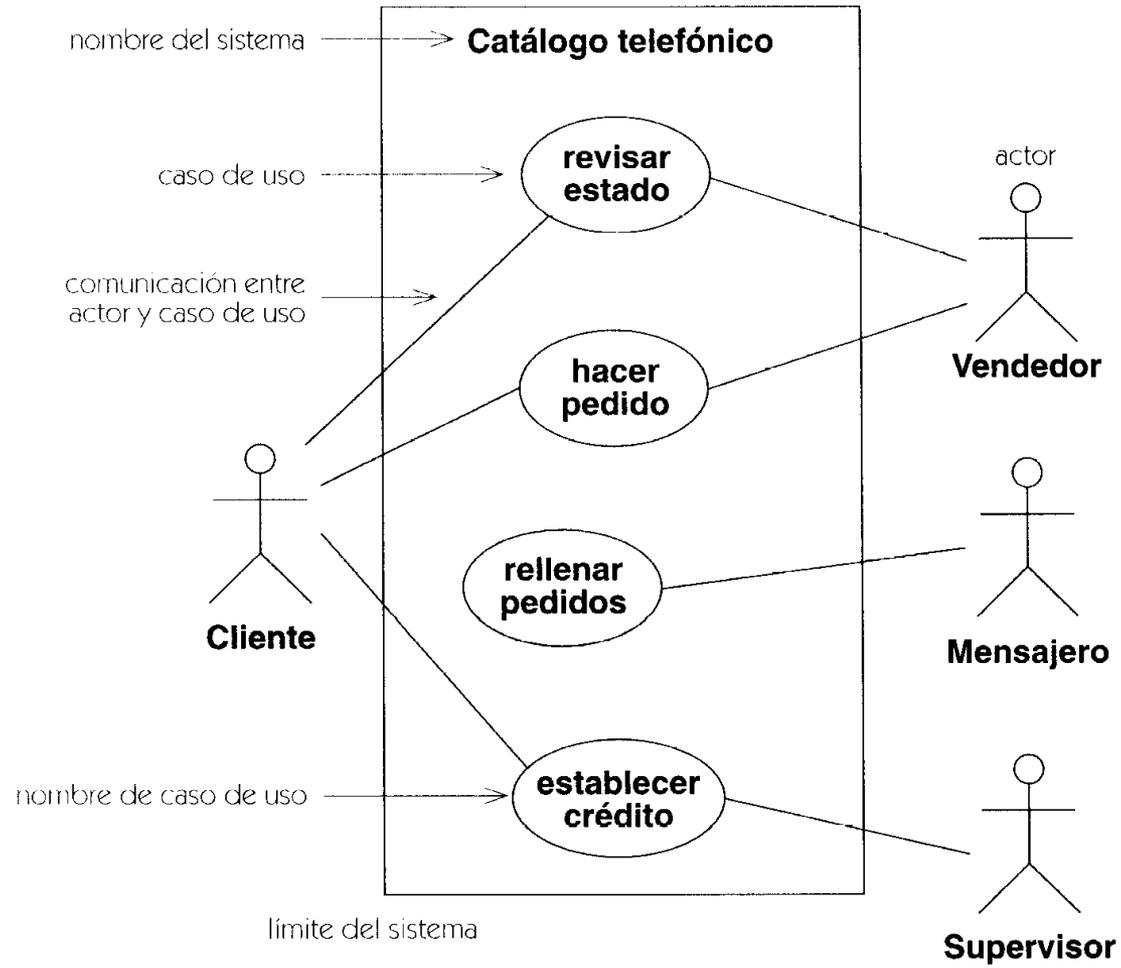
DIAGRAMA DE CASOS DE USO

Cuando utilizar <<extends>> o <<include>>

- Usar relaciones extendidas para comportamientos excepcionales, opcionales o que rara vez suceden.
- Usar relaciones de inclusión para comportamientos que se comparten entre dos o más casos de uso



DIAGRAMA DE CASOS DE USO



IDENTIFICACIÓN DE CLASES

Un objeto potencial debe satisfacer las siguientes características para ser considerado posible miembro del modelo:

- **Retener Información:** Se requiere guardar información para el funcionamiento del sistema
- **Servicios necesarios:** Conjunto de operaciones identificables que permitan cambiar el estado de sus atributos.
- **Múltiples atributos.**
- **Atributos comunes:** El conjunto de atributos de la clase debe aplicar a todas las ocurrencias del objeto
- **Operaciones comunes:** El conjunto de operaciones debe aplicar a todos los objetos de la clase.



IDENTIFICACIÓN DE CLASES

Clase Dispositivo: Entidades externas (sensores, motores, teclado).

Clase Propiedad: Representa alguna propiedad importante del entorno (Ejemplo: Establecimiento de créditos dentro del contexto de una aplicación de préstamos hipotecarios)

Clase interacción: Modelan interacciones entre objetos. (Ejemplo: Reservación de libros)



DIAGRAMA DE CLASES

¿Qué es ?

Modela los conceptos del dominio de la aplicación: describe el sistema identificando sus clases y relaciones.

Componentes:

- ❑ **Clases:** Atributos, Métodos, Visibilidad.
- ❑ **Relaciones:** Herencia, Composición, Agregación, Asociación y uso.



DIAGRAMA DE CLASES

Clases

Representada por un rectángulo con tres divisiones:

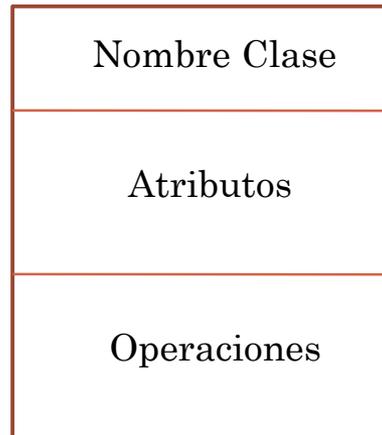


DIAGRAMA DE CLASES

Atributos

Pueden ser de tres tipos (definen el grado de comunicación y la visibilidad de los atributos con el entorno).

- ❑ **Public (+):** Indica que el atributo será visible tanto dentro como fuera de la clase.
- ❑ **Private (-):** Indica que el atributo sólo será accesible desde la clase.
- ❑ **Protected(#):** Indica que el atributo no será accesible desde fuera de la clase, pero si podrá ser accedido por métodos de la clase y subclases que se derivan (Herencia).



DIAGRAMA DE CLASES

Métodos

Presentan tres características:

- ❑ **Public (+):** Indica que el métodos será visible tanto dentro como fuera de la clase.
- ❑ **Private (-):** Indica que el método sólo será accesible desde la clase.
- ❑ **Protected(#):** Indica que el método no será accesible desde fuera de la clase, pero si podrá ser accedido por métodos de la clase y subclases que se derivan (Herencia).



DIAGRAMA DE CLASES

Relaciones entre Clases

- ❑ **Asociación:** Representa la relación existente entre dos clases de forma conceptual. Se documenta destacando el rol que juega cada clase en la asociación.

Simbología: _____

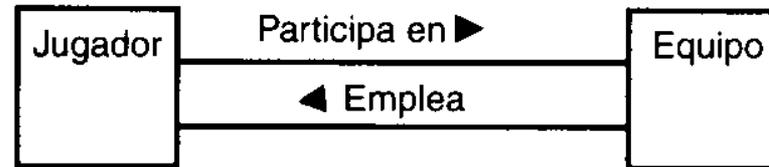
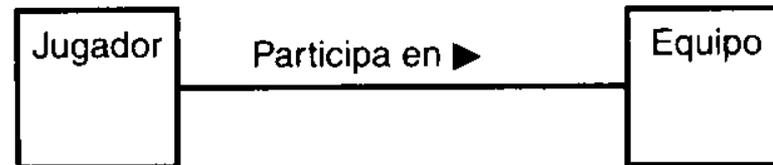
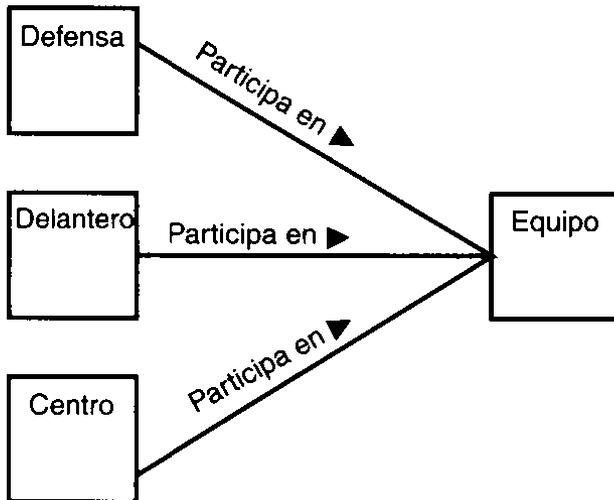


DIAGRAMA DE CLASES

Restricciones en las asociaciones

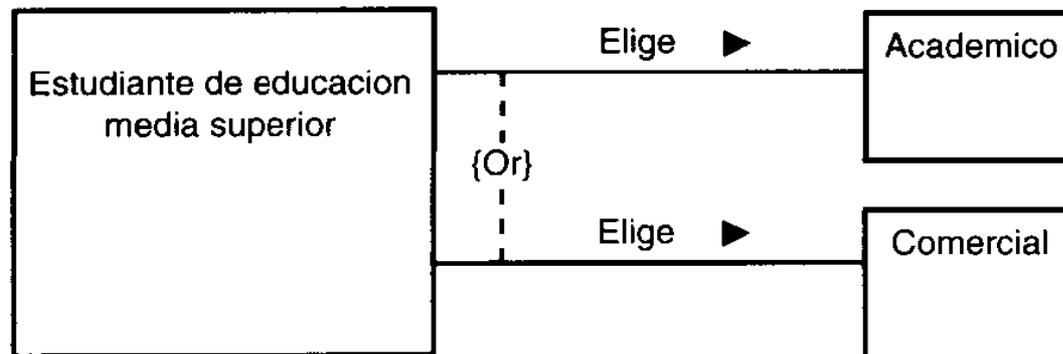
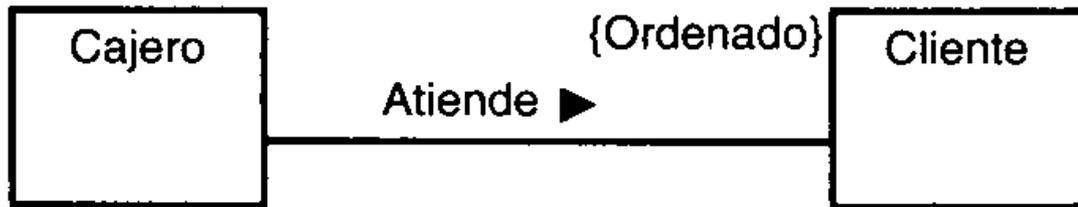


DIAGRAMA DE CLASES

Relaciones entre Clases

Cardinalidad (Multiplicidad):

- Indica el grado y nivel de dependencia entre clases.
- Cantidad de objetos de una clase que pueden relacionarse con un objeto de una clase asociada,

Puede ser:

- **1 (uno y sólo uno)**
- **0...1 (cero o uno)**
- **N...M (desde N hasta M)**
- *** (cero o varios)**
- **0...* (cero o varios)**
- **1...* (uno o varios)**

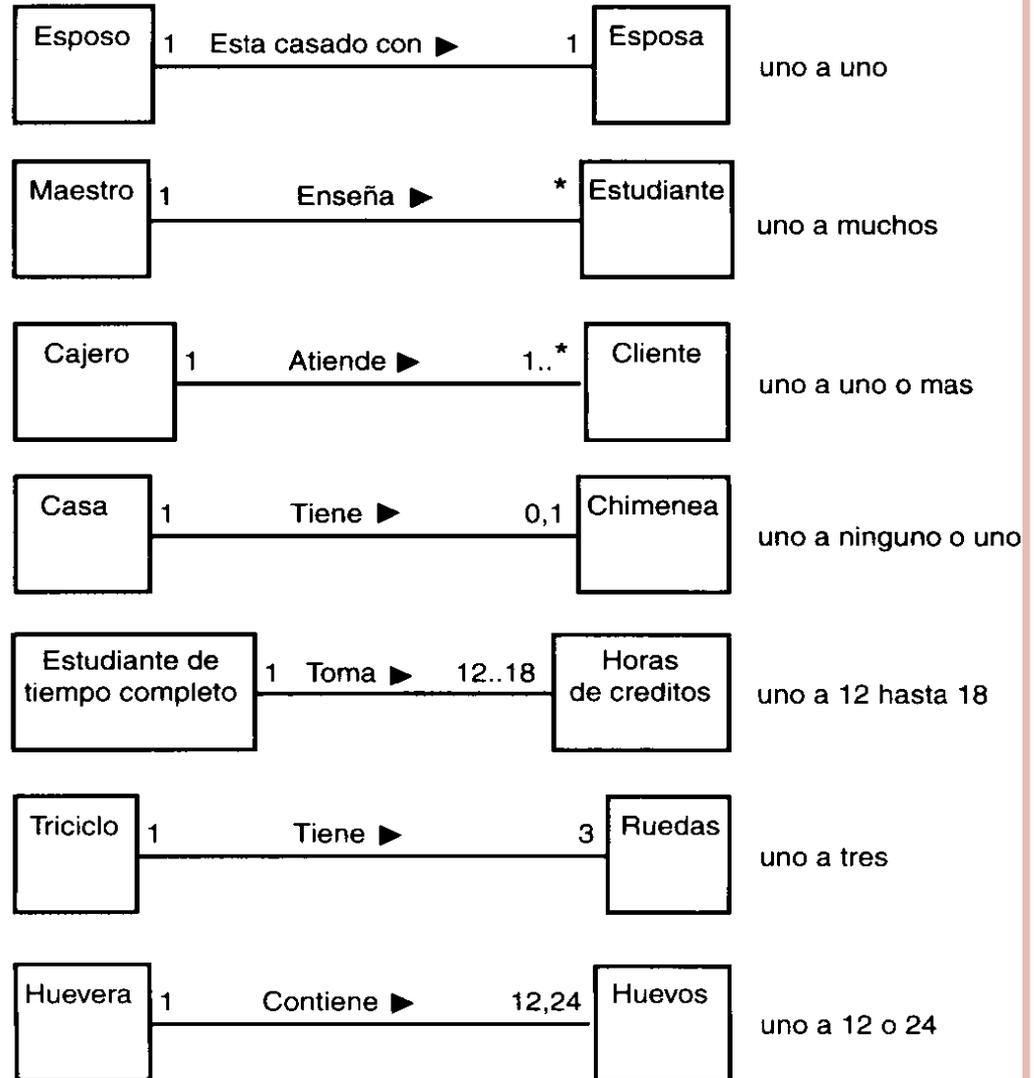


DIAGRAMA DE CLASES

Asociaciones Reflexivas

- Una clase es asociación consigo misma
- Una clase tiene objetos que pueden jugar diversos papeles o roles.

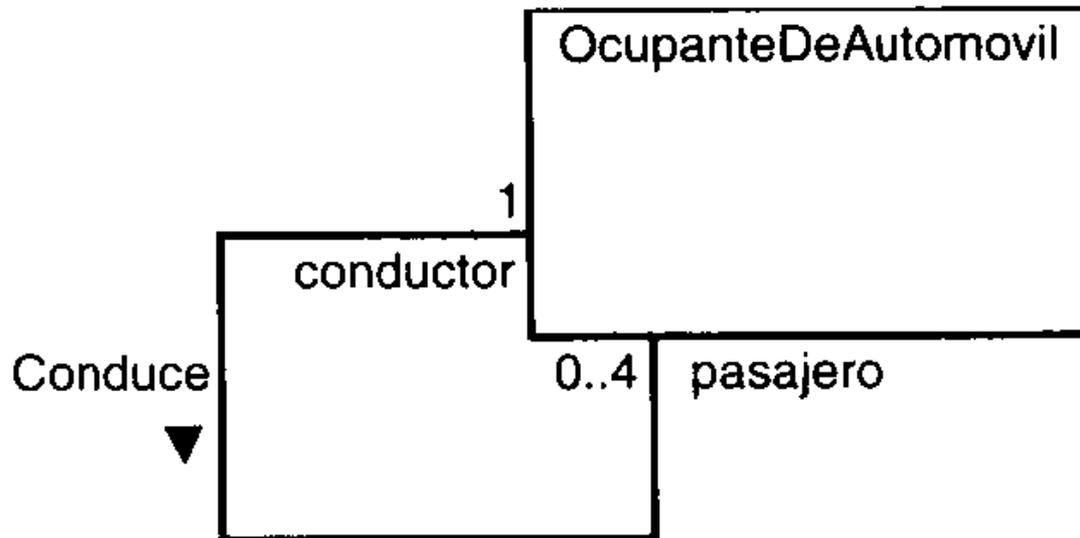


DIAGRAMA DE CLASES

Relaciones entre Clases

□ Herencia (especialización/generalización):

Indica que una subclase hereda métodos y atributos de una superclase.

La subclase además de poseer sus propios métodos y atributos, poseerá las características visibles de la superclase (Public y Protected).

Simbología

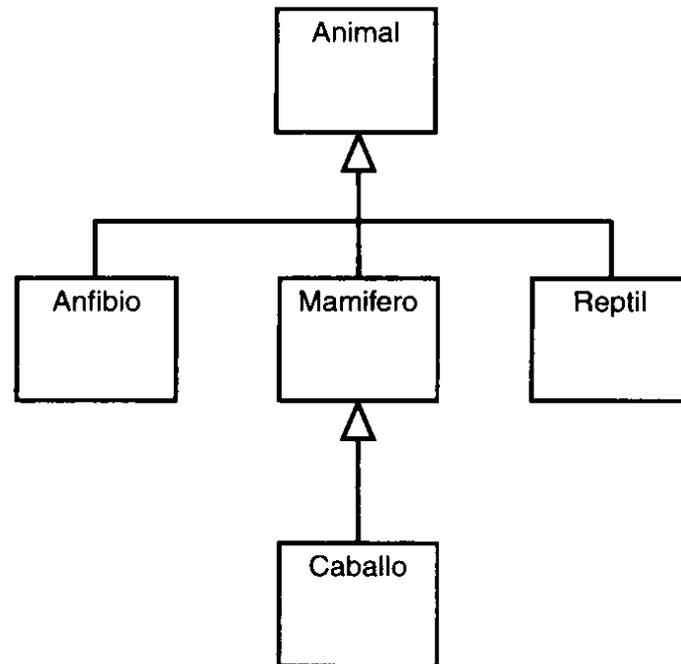


DIAGRAMA DE CLASES

Relaciones entre Clases: Clases abstractas

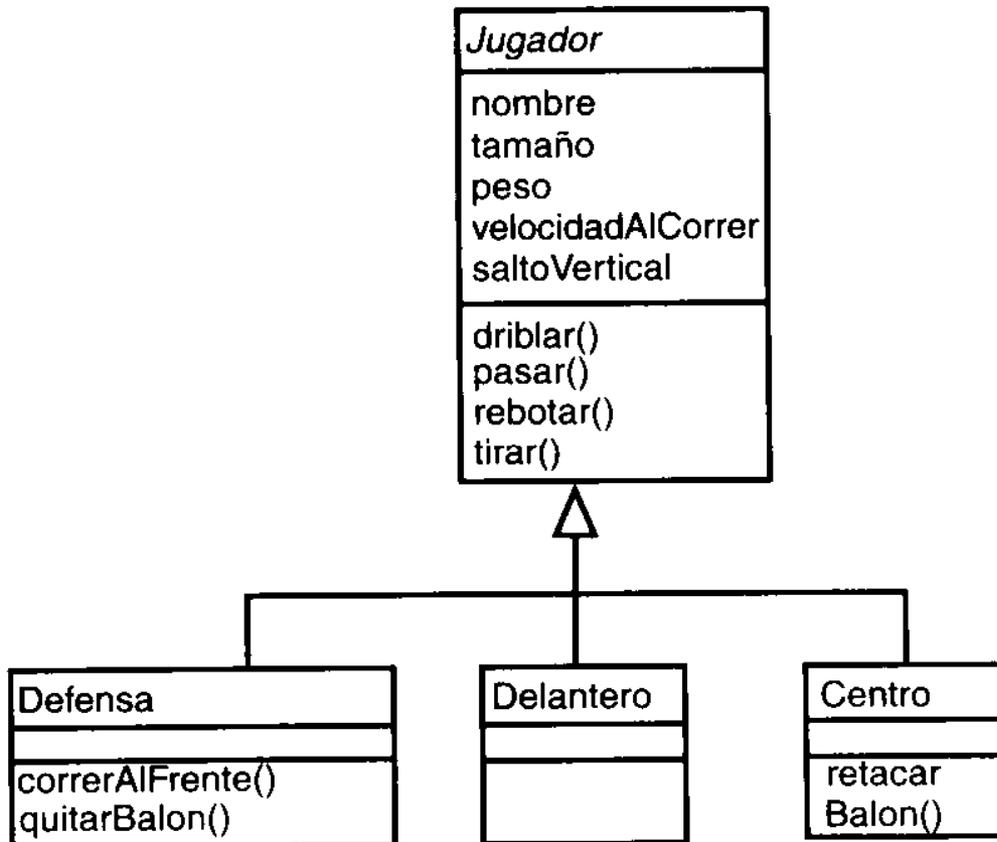


DIAGRAMA DE CLASES

Relaciones entre Clases

■ Agregación:

Se da cuando una clase describe objetos que agregan otros objetos.
(Todo Parte)

Simbología

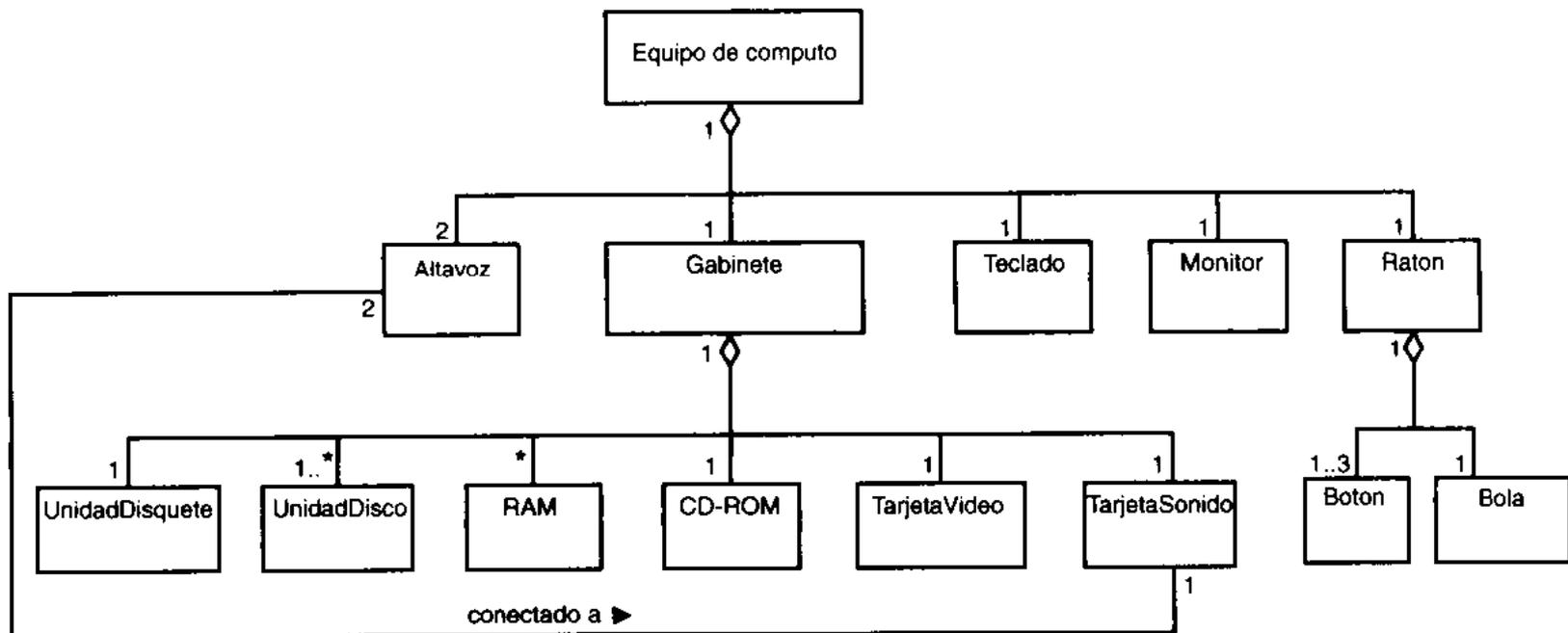


DIAGRAMA DE CLASES

Relaciones entre Clases

❑ Restricciones en las agregaciones:

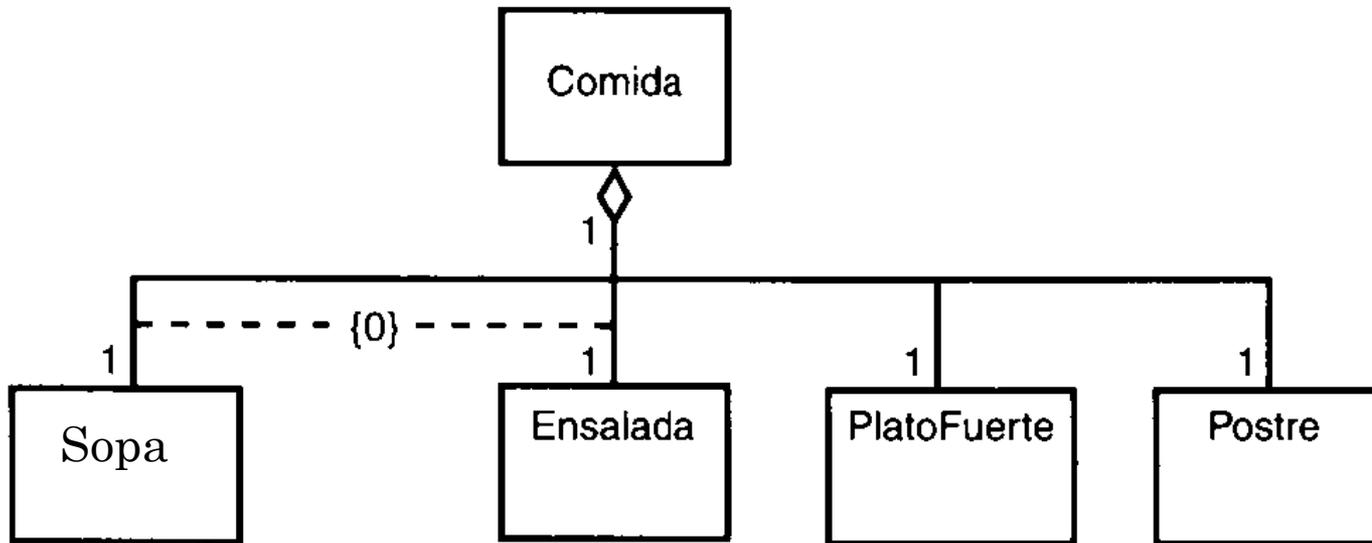


DIAGRAMA DE CLASES

Relaciones entre Clases

- ❑ **Composición (Agregación Exclusiva):** Se da cuando un objeto contiene un numero de otros objetos y cuando el objeto contenedor se eliminan todas las instancias de los objetos que están contenidos desaparecen.

Es una asociación fuerte, que implica tres cosas:

- **Dependencia existencial:** El elemento dependiente desaparece al destruirse el que lo contiene.
- **Hay una pertenencia fuerte:** Se puede decir que el objeto contenido es parte constitutiva y vital del que lo contiene
- Los objetos contenidos no son compartidos, esto es, no hacen parte del estado de otro objeto.

Simbología



DIAGRAMA DE CLASES

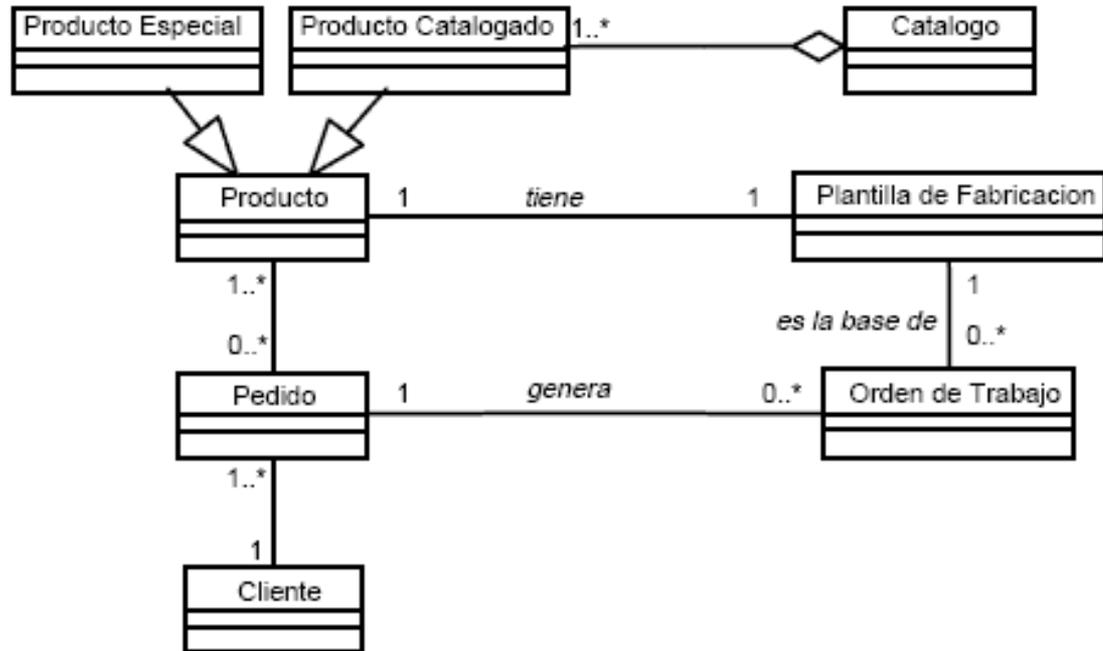


DIAGRAMA DE SECUENCIA

¿Qué es un diagrama de secuencia?

- ❑ Modela la secuencia lógica, a través del tiempo, de los mensajes entre las instancias (objetos).

Notación:

- ❑ **Objetos:** Se colocan en la parte superior del diagrama.
- ❑ **Línea de tiempo:** Representado por una línea discontinua. -----
 - ❑ Avanza de arriba hacia abajo.
- ❑ **Línea de vida de los objetos:** Representado con un rectángulo. 
 - ❑ Desciende de cada uno de los objetos.
 - ❑ Indica la activación de un objeto (ejecución de una de las operaciones del objeto).



DIAGRAMA DE SECUENCIA

Notación:

❑ Mensajes :

- ❑ Representados con flechas. 
- ❑ Conectan una línea de vida con otra.
- ❑ Su ubicación en la dimensión vertical indica el momento en el que sucede dentro de la secuencia. Los más cercanos a la parte superior del diagrama se ejecutan primero, los que ocurren después cerca de la parte inferior.

- ❑ **Recursividad:** Representada con una flecha que regresa al objeto que inicio recursividad. Se da cuando el objeto ejecuta una de sus operaciones.

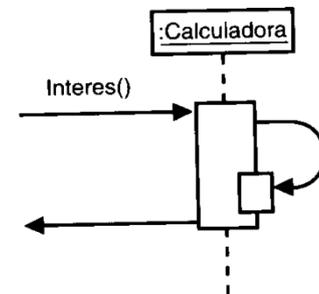


DIAGRAMA DE SECUENCIA

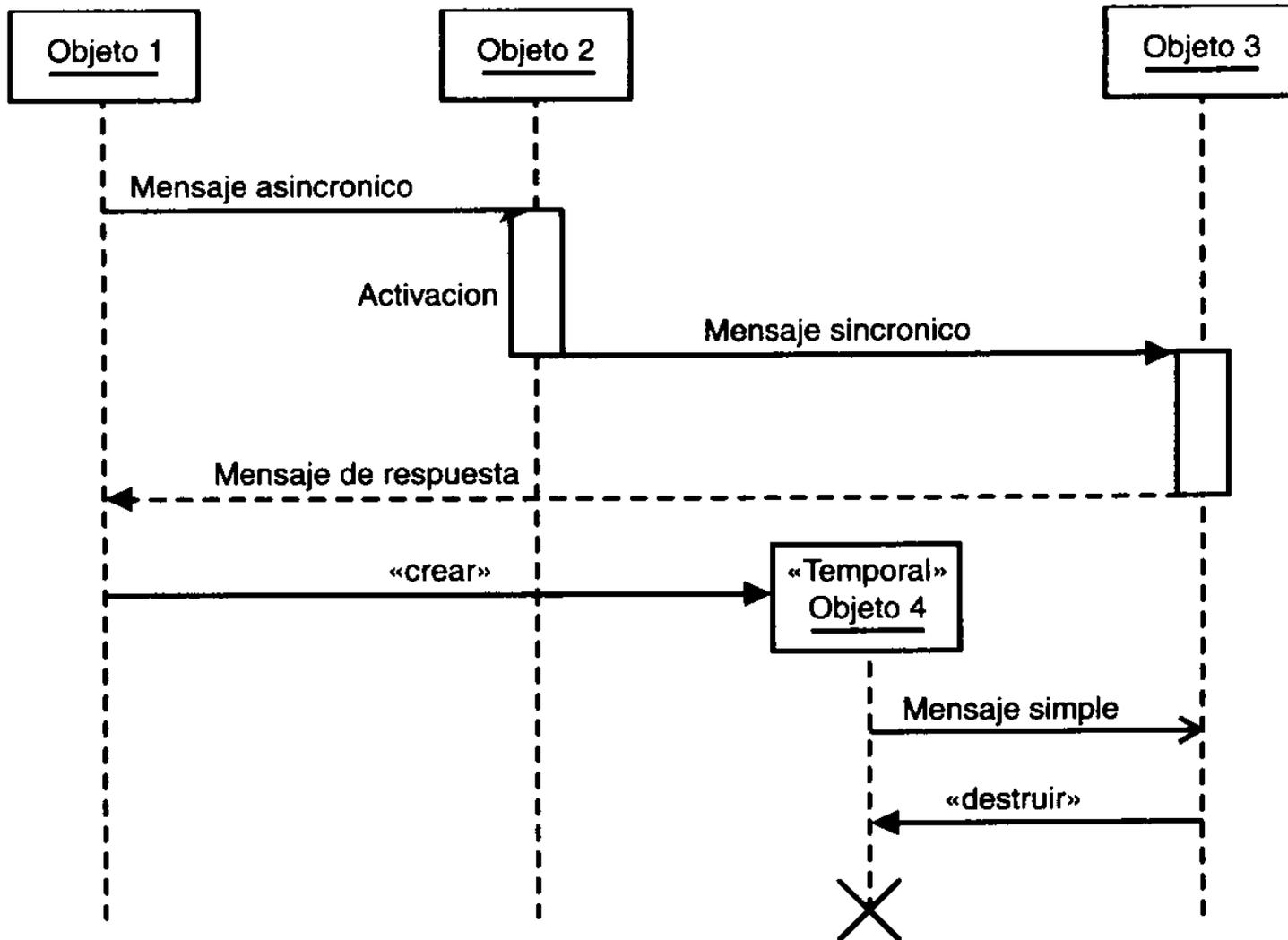


DIAGRAMA DE COLABORACIÓN

¿Qué es un Diagrama de Colaboración?

- ❑ Un Diagrama de Colaboración describe las interacciones entre objetos haciendo énfasis en la estructura de la colaboración
- ❑ Muestra la relación entre objetos y los mensajes que se envían entre sí.

Notación

- ❑ **Asociación:** Se representa con una línea continua. 
- ❑ **Mensaje:** Se representa con una flecha cerca de la de la línea de asociación. La flecha apunta al objeto receptor. 
- ❑ **Tipo de Mensaje:** Se muestra en una etiqueta cerca de la flecha. Indica al objeto receptor que ejecute una de sus operaciones. Finalizará con paréntesis, estos contienen los parámetros (en caso de que se requieran) con los que funcionara la operación. Ej: Agregar(Par1,Par2...)
- ❑ **Secuencia del Mensaje:** Representada por una cifra que se agrega a la etiqueta del mensaje. La cifra y el mensaje se separan por medio de dos puntos (:). Ej: 1:Agregar(Par1, Par2).

DIAGRAMA DE COLABORACIÓN

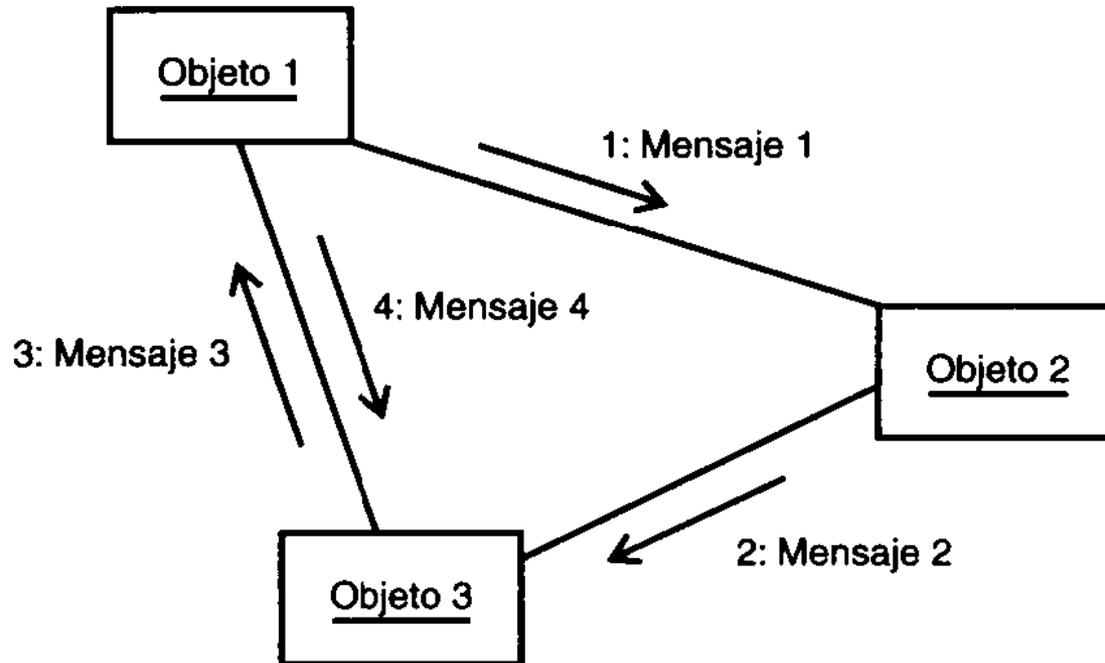
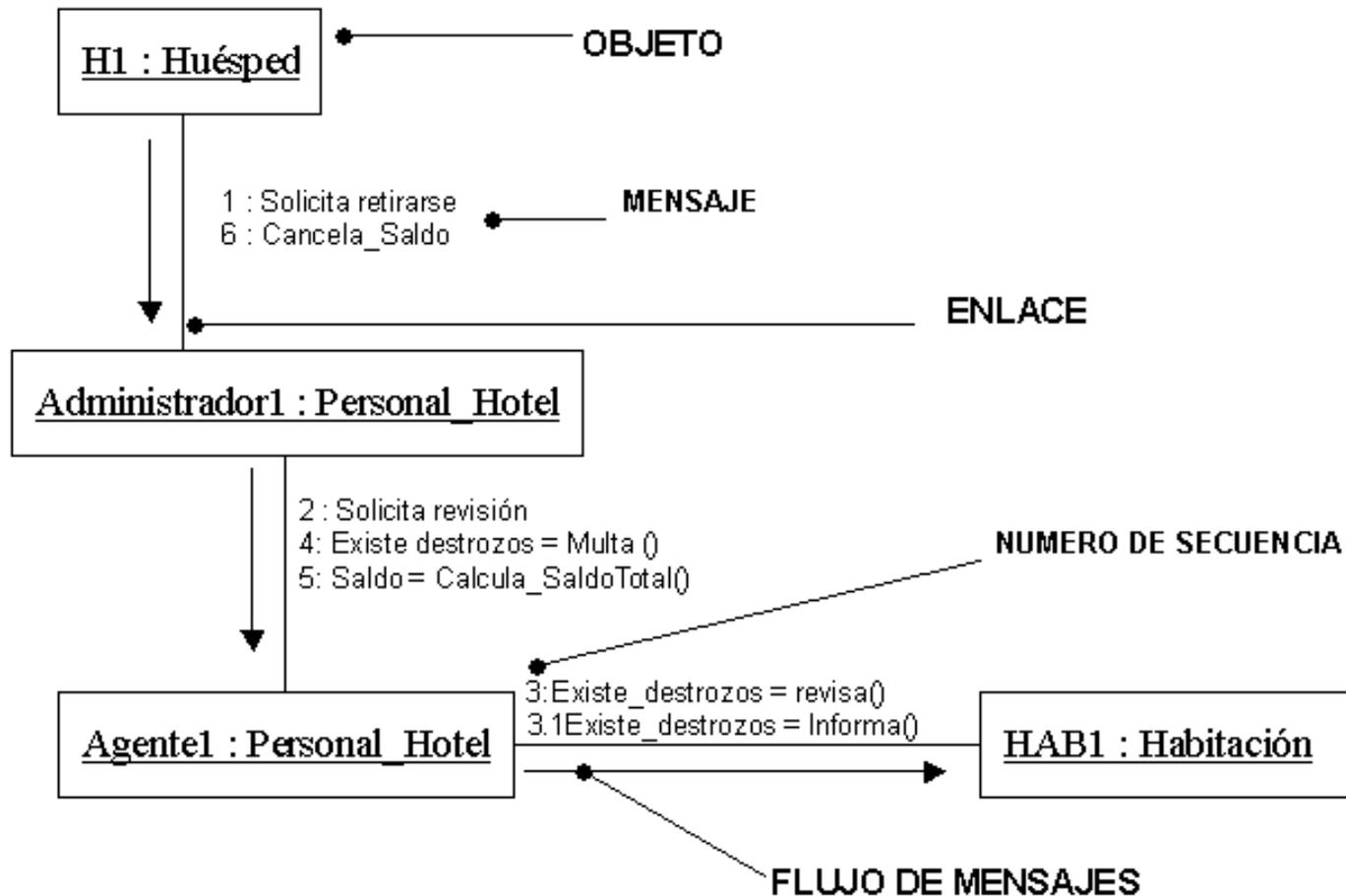
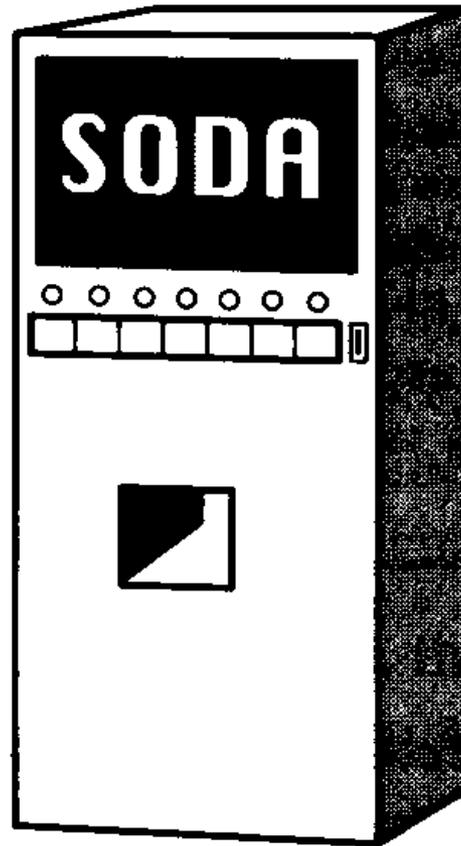
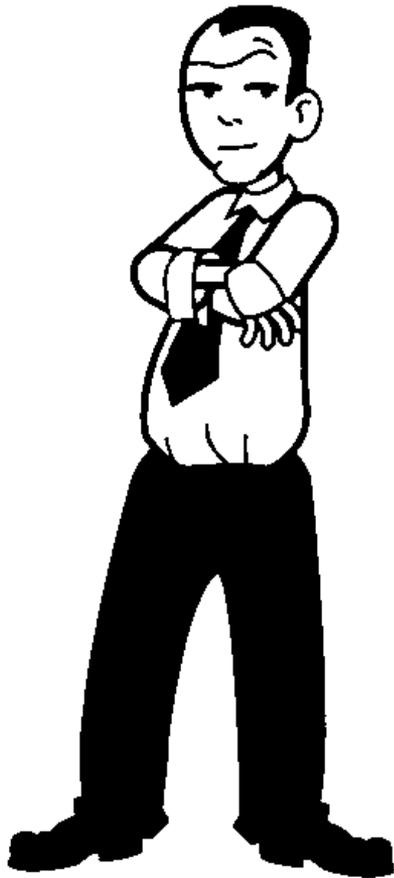


DIAGRAMA DE COLABORACIÓN



EJEMPLO UML



**MÁQUINA
DISPENSADORA
DE
GASEOSAS**



CASOS DE USO

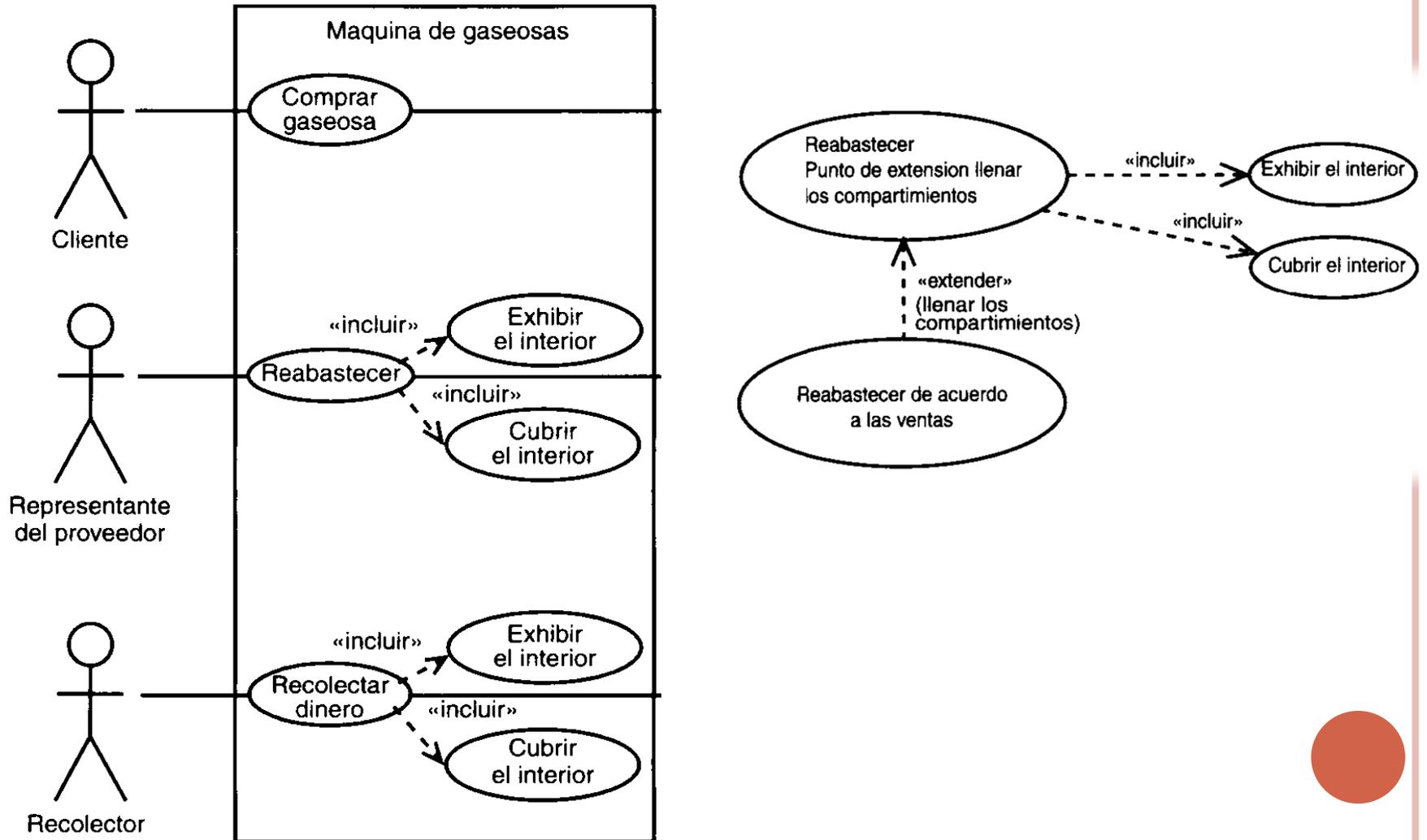


DIAGRAMA DE SECUENCIA

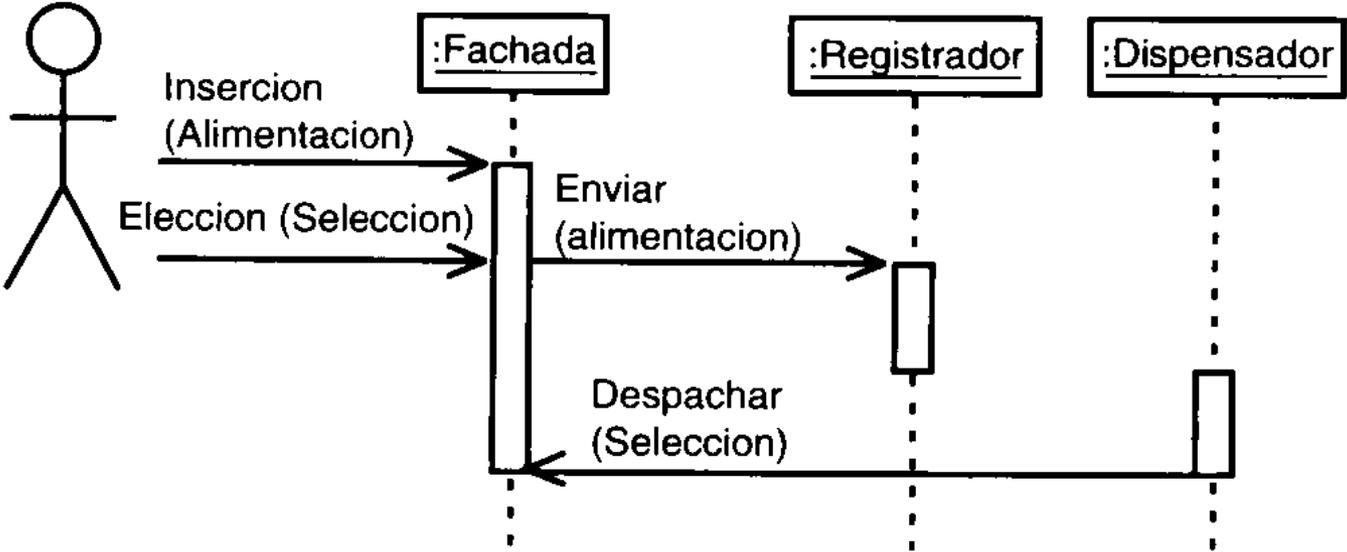


DIAGRAMA DE COLABORACION MEJOR DE LOS CASOS

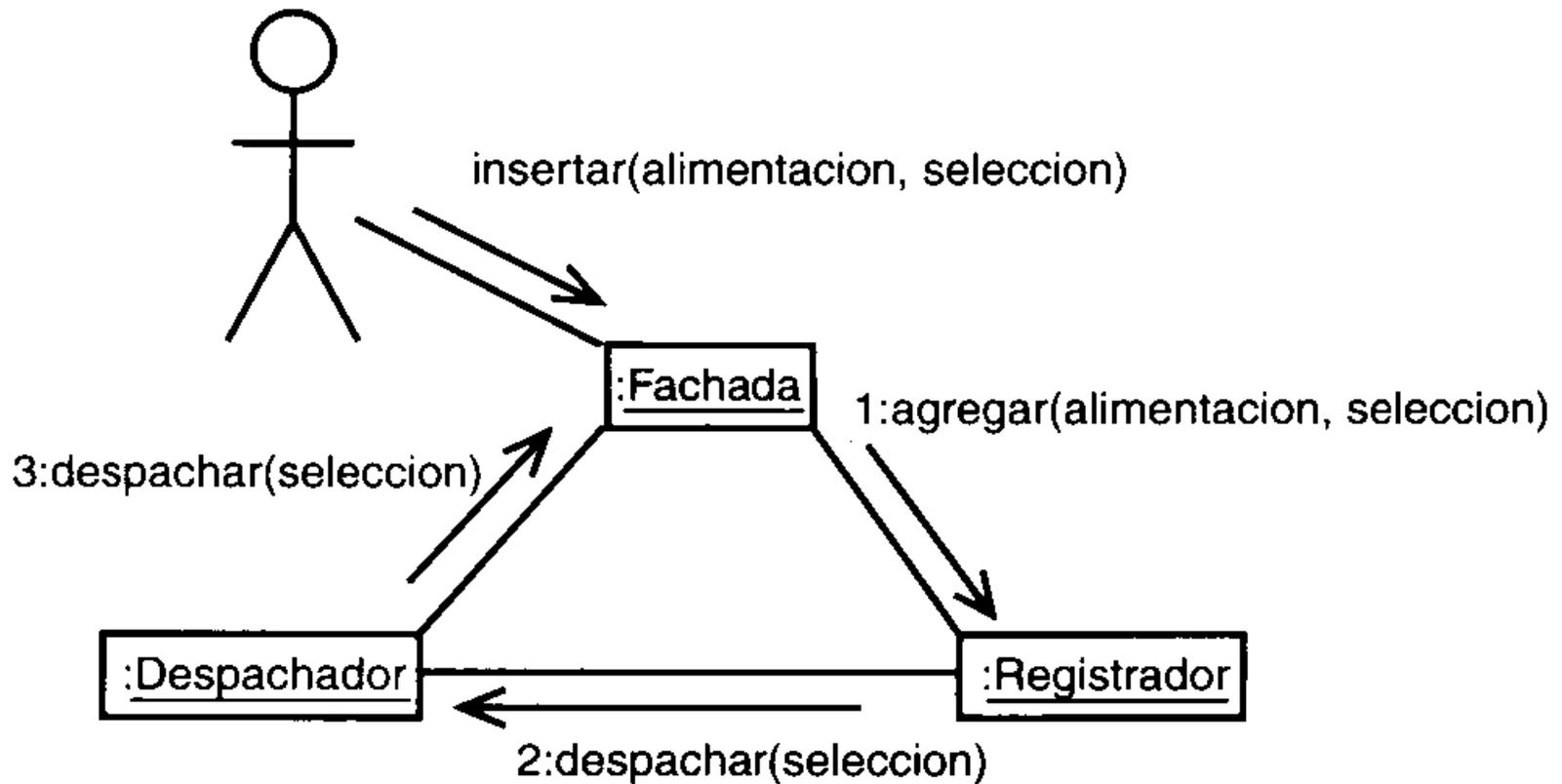


DIAGRAMA DE SECUENCIA GENÉRICO

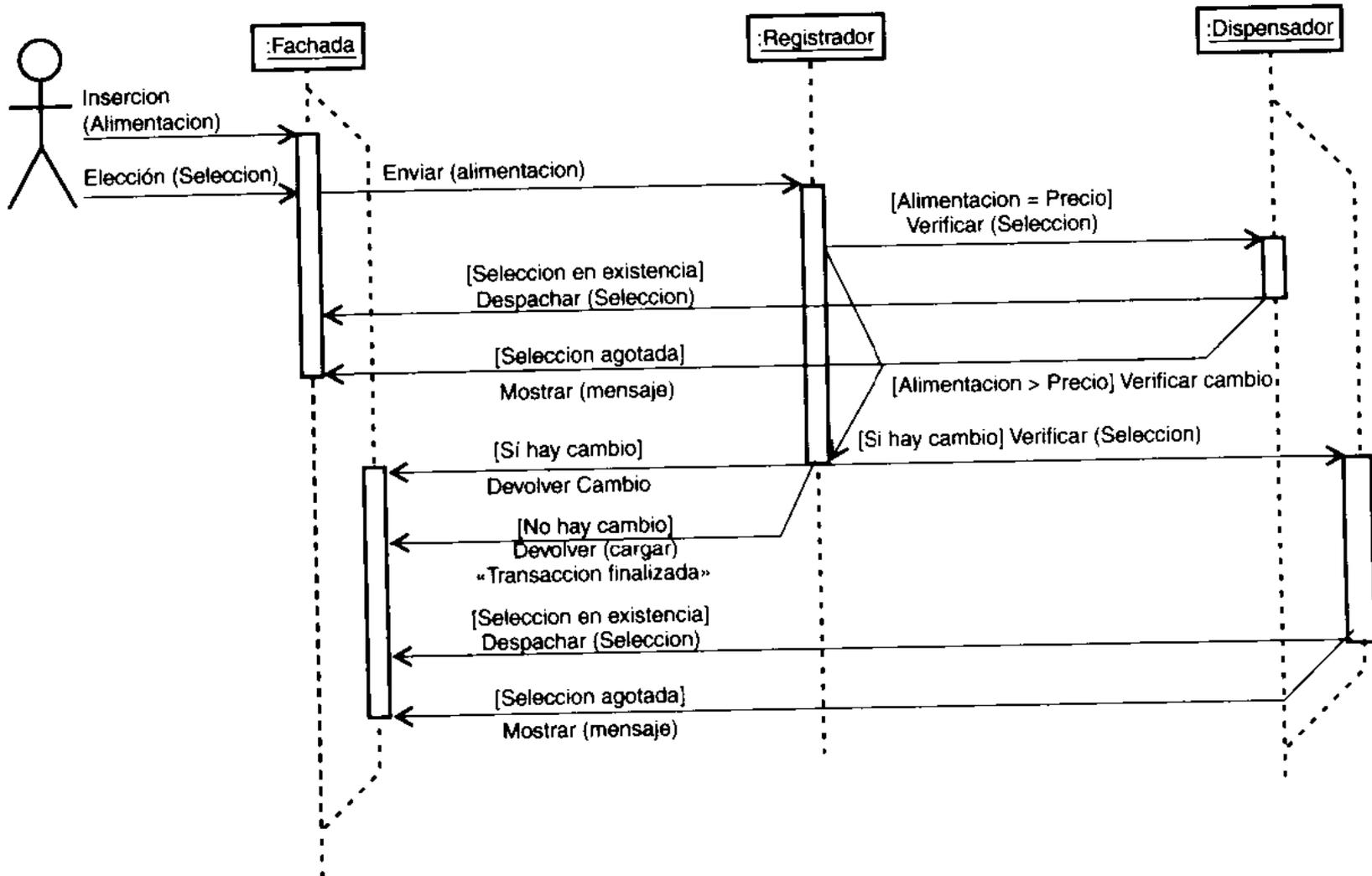


DIAGRAMA DE COLABORACION GENÉRICO

