



INTRODUCCION AL UML

(Lenguaje Unificado de Construcción de Modelos)

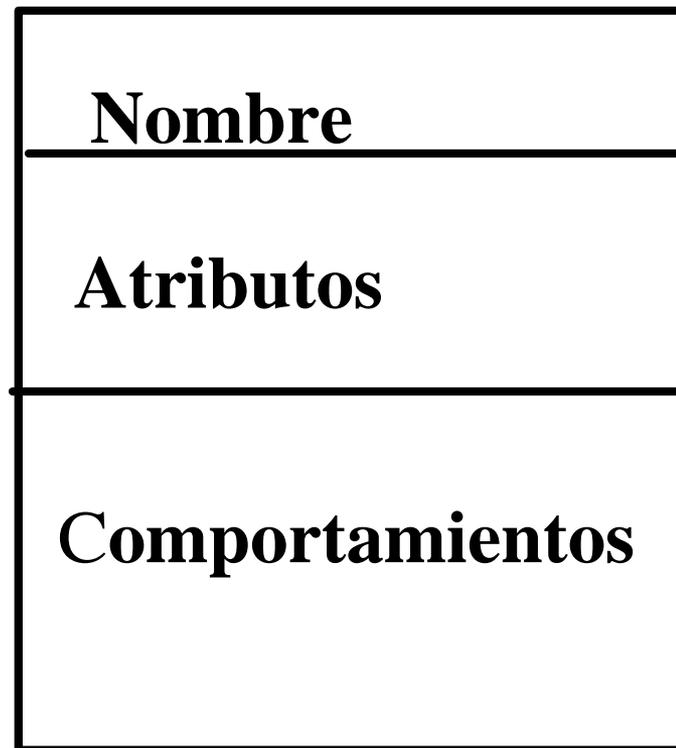


UML

- Es una herramienta que nos permitirá expresarnos en un lenguaje común
- Permite facilitar la comunicación entre las distintas áreas de una organización

UML: la Orientación a Objetos

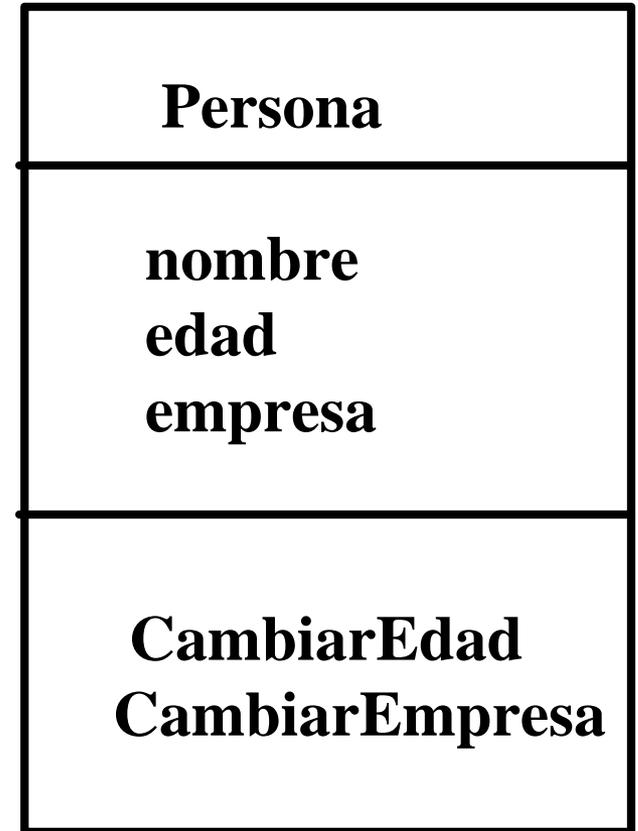
Un objeto es un concepto (personas, cosas, hechos, ideas, etc)



Atributo y comportamiento

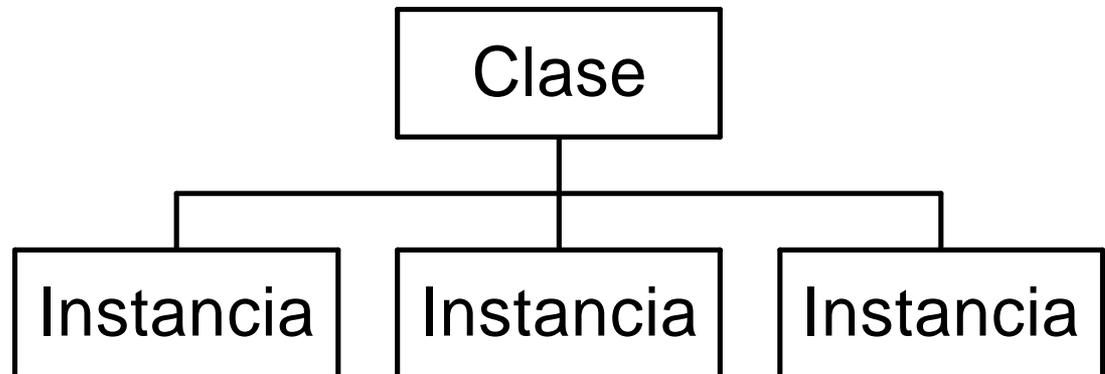
- **Atributo**: son las características o cualidades del objeto (también se denominan propiedades)
- **Comportamiento**: son las acciones, aquello que el objeto sabe o puede hacer

Ejemplo de Objeto



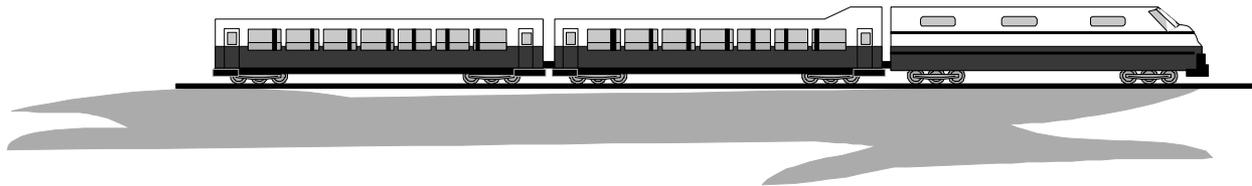
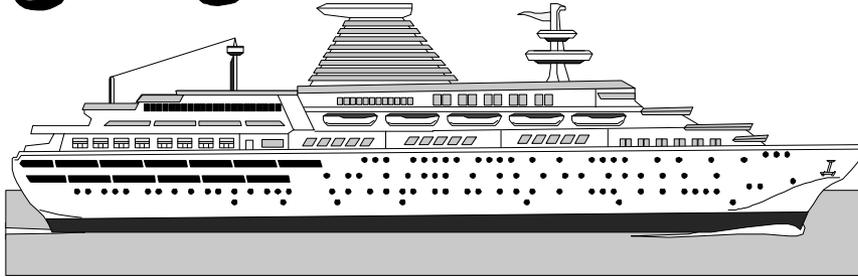
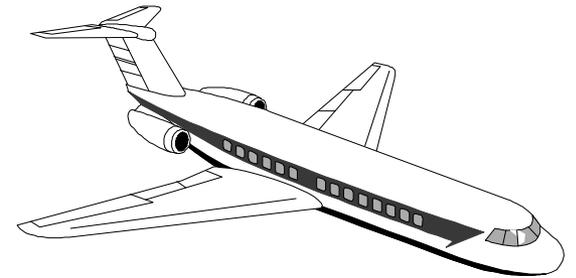
Noción de Clase e Instancia

- Todos los objetos naturalmente se agrupan en *categorias* (clases)
- Los objetos que están comprendidos dentro de las clases se denominan instancias



Actividades:

1. Identifique una clase que agrupe todos estos objetos
2. Agrupe diversos objetos en distintas clases.



Instancias



Instancia persona n° 1
-nombre = SALAS
-edad=35
-empresa=IPV

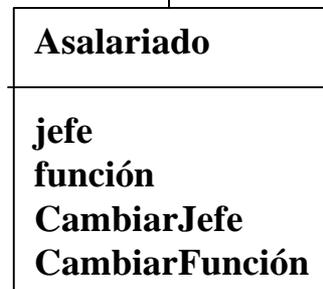
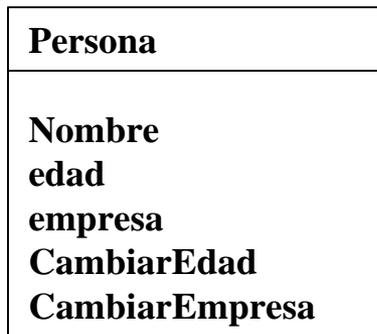
Instanciación



Instancia persona n° 2
-nombre = FUNES
-edad=55
-empresa=VPI

Jerarquía de Clases y herencia

- El mecanismo de la herencia permite definir nuevas clases a partir de clases existentes



Instancia



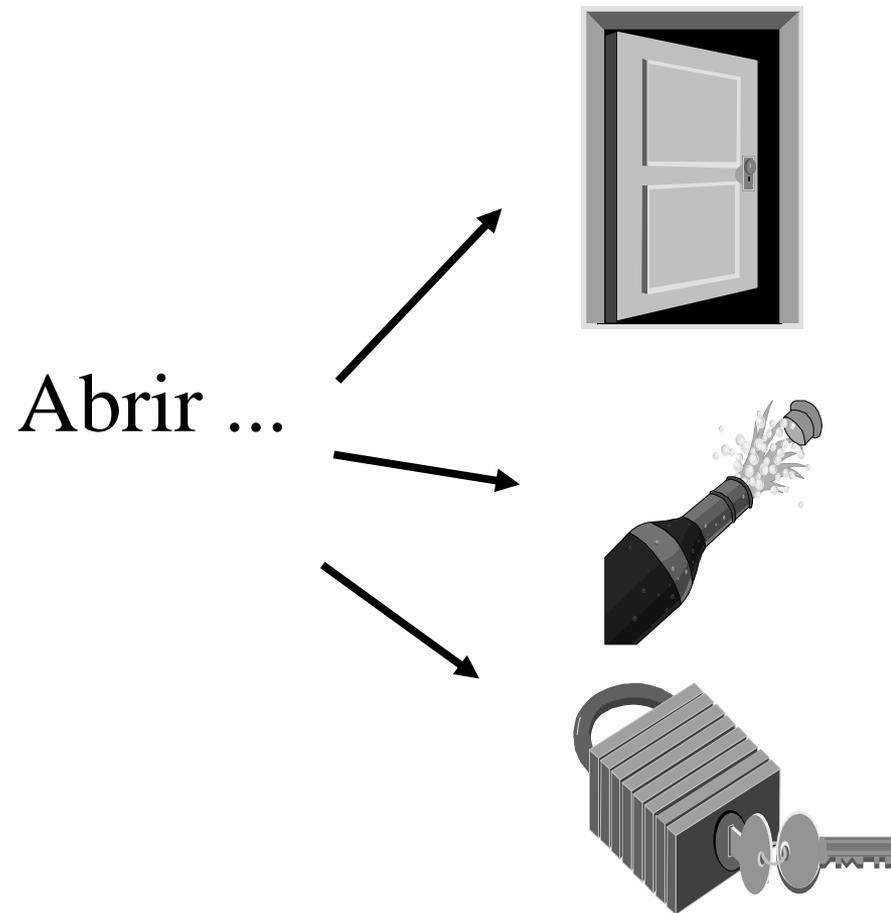
Instancia de persona n° 1
-nombre = RODRIGUEZ
-edad=36
-empresa=MUNI
-jefe=SANENZ
-función=encargado sección

Polimorfismo

- El polimorfismo es una característica de la OO (orientación a objetos) que permite redefinir un comportamiento (método) heredado por una superclase

Polimorfismo

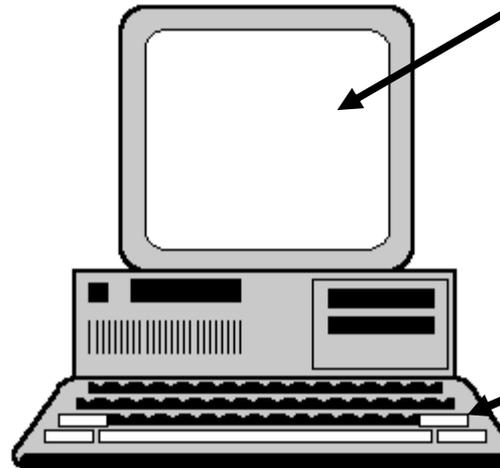
El polimorfismo permite usar los mismos términos del cliente.



Encapsulamiento e Interfaces

- ¿Cómo funciona?

- ¿A quién le importa!



Pantalla

Teclado

Modelo Conceptual

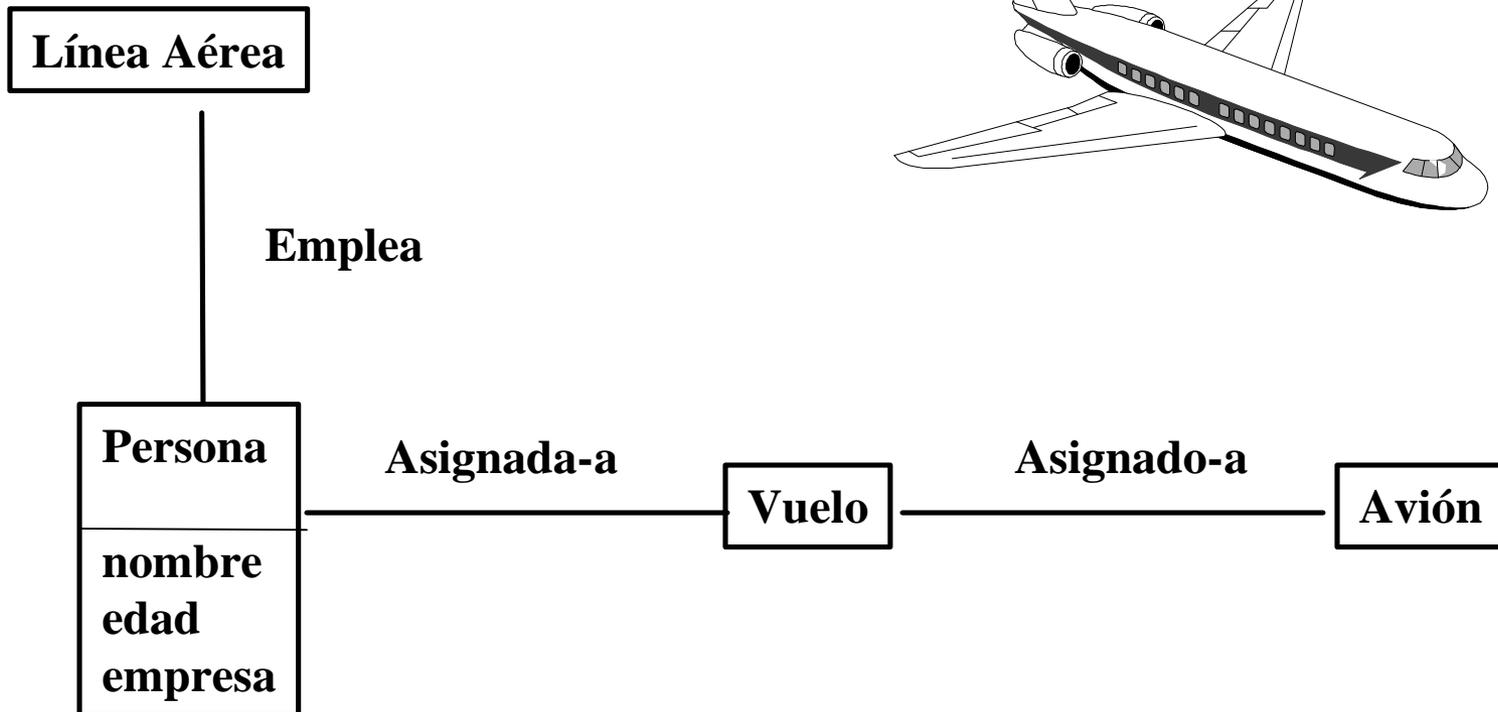
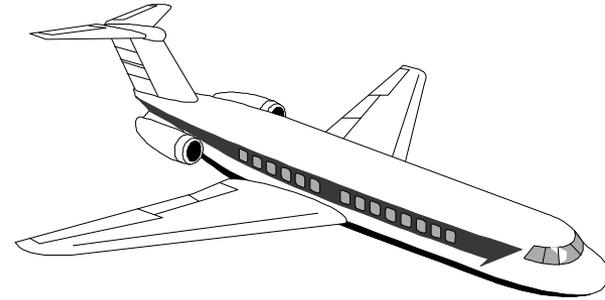
- **Identifica los objetos.**
- **Representa cosas del mundo real.**
- **Es un diagrama estático donde no se define ninguna operación (proceso).**
- **Ayuda a esclarecer la terminología.**

Es el artefacto más importante en la etapa del análisis del problema.

Modelo Conceptual

- **Nos muestra:**
 - **Clases**
 - **Asociaciones entre esas Clases**
 - **Atributos de dichas Clases**

Ejemplo



Maneras de definir Clases



Por el *Símbolo*

“Una venta representa el evento de una transacción de compra. Tiene fecha y hora”

Definición *intensiva*

Venta-1
Venta-2 Venta-3
Venta-4

Definición *extensiva*

La asignación de nombres

- **Se puede aplicar la metodología del cartógrafo:**
 - **Utilizar los nombres existentes en el territorio.**
 - **Excluir las características irrelevantes.**
 - **No agregar cosas que no existan.**

Descomposición del problema

- **Ante los problemas complejos**
 - “divide y vencerás”
- **Dividimos el problema en partes comprensibles**
- **Conviene llevarla a cabo a partir de las clases**

Descomposición del problema (cont.)

- **Una guía para esta fase:**
 - **Identificar varias clases**
 - **Documentar los resultados en un modelo conceptual**

Clases del Caso de la Caja

Local

Caja

Venta

Agreguemos otras clases que puedan identificar:

Estrategias para identificar las clases

- **A partir de una lista de categorías de clases**
- **Identificación de frases nominales**

Identificación de frases nominales

Acción del actor

Respuesta del sistema

1. Este caso comienza cuando un Cliente llega a una caja con productos que desea comprar

2. El Cajero registra el identificador de cada producto.

Si hay varios productos de una misma categoría, el Cajero también puede introducir la cantidad

3. Determina el precio del producto e incorpora a la transacción actual la información correspondiente.

Se presenta la descripción y el precio del producto actual.

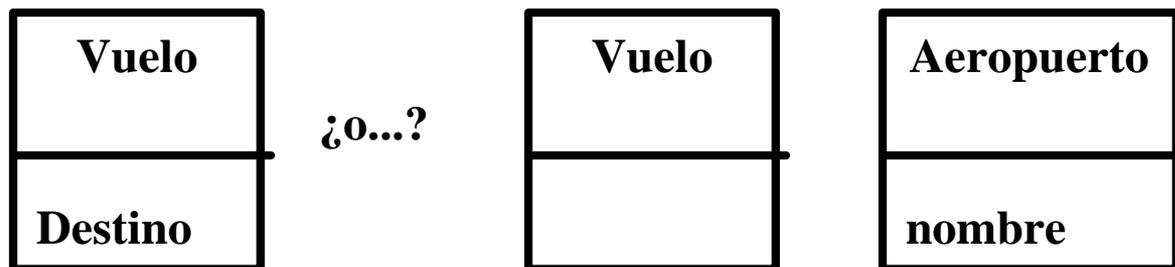
Aplicación

- **Usando la lista de categorías de clases y análisis de frases nominales, construyamos una lista de clases de una aplicación del Video Club o la Biblioteca.**

Identificando las clases

A veces confundimos clases y atributos.

Si consideramos algo como atributo (que no es un número o texto en el mundo real), probablemente éste sea un objeto y no un atributo.



En caso de duda, convertir el atributo en clase.

Resumiendo

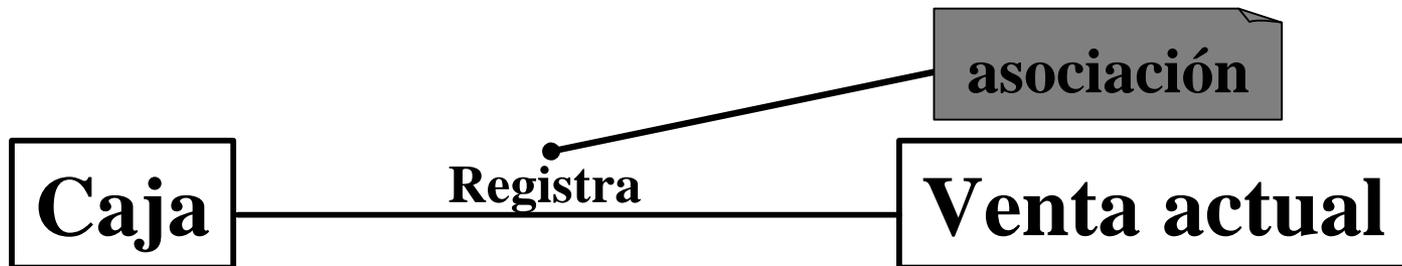
Clase “es una descripción de un conjunto de objetos que comparten los mismos atributos, relaciones y comportamientos”

En UML las asociaciones son relaciones entre las clases

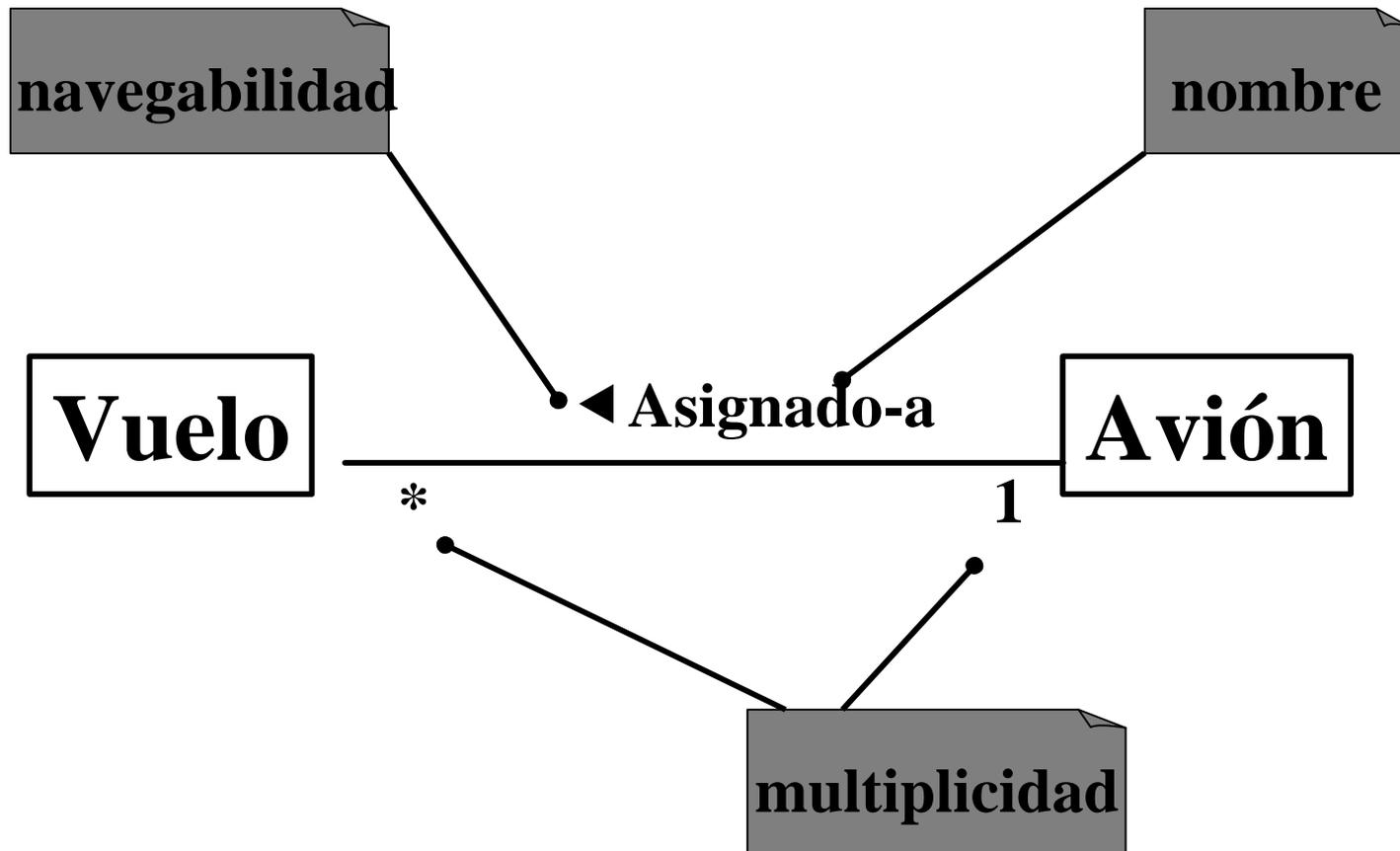


Asociaciones

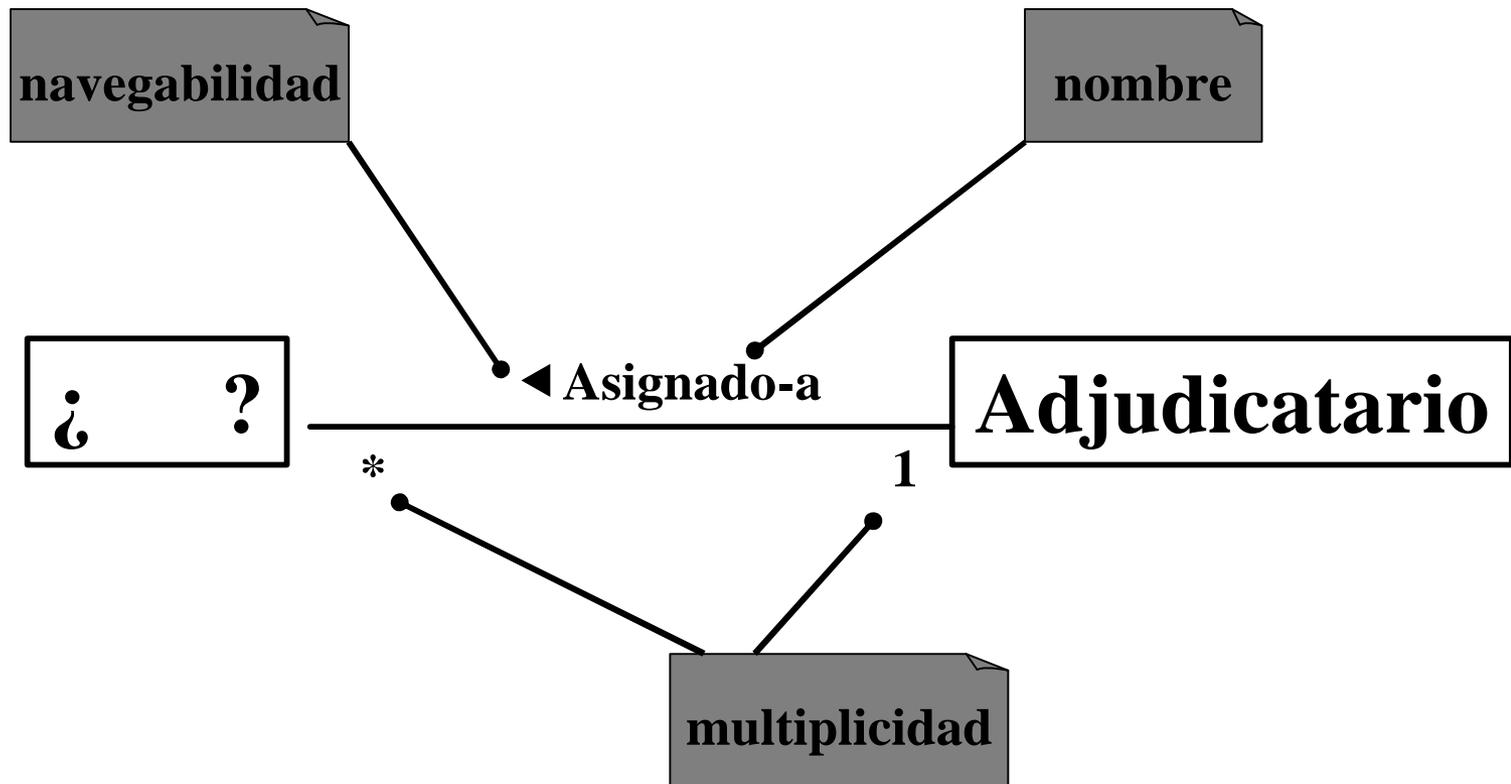
- **La asociación es una relación entre dos clases que indica alguna conexión significativa e interesante entre ellas**



Notación de las asociaciones en UML



Notación de las asociaciones en UML (Ejemplo)

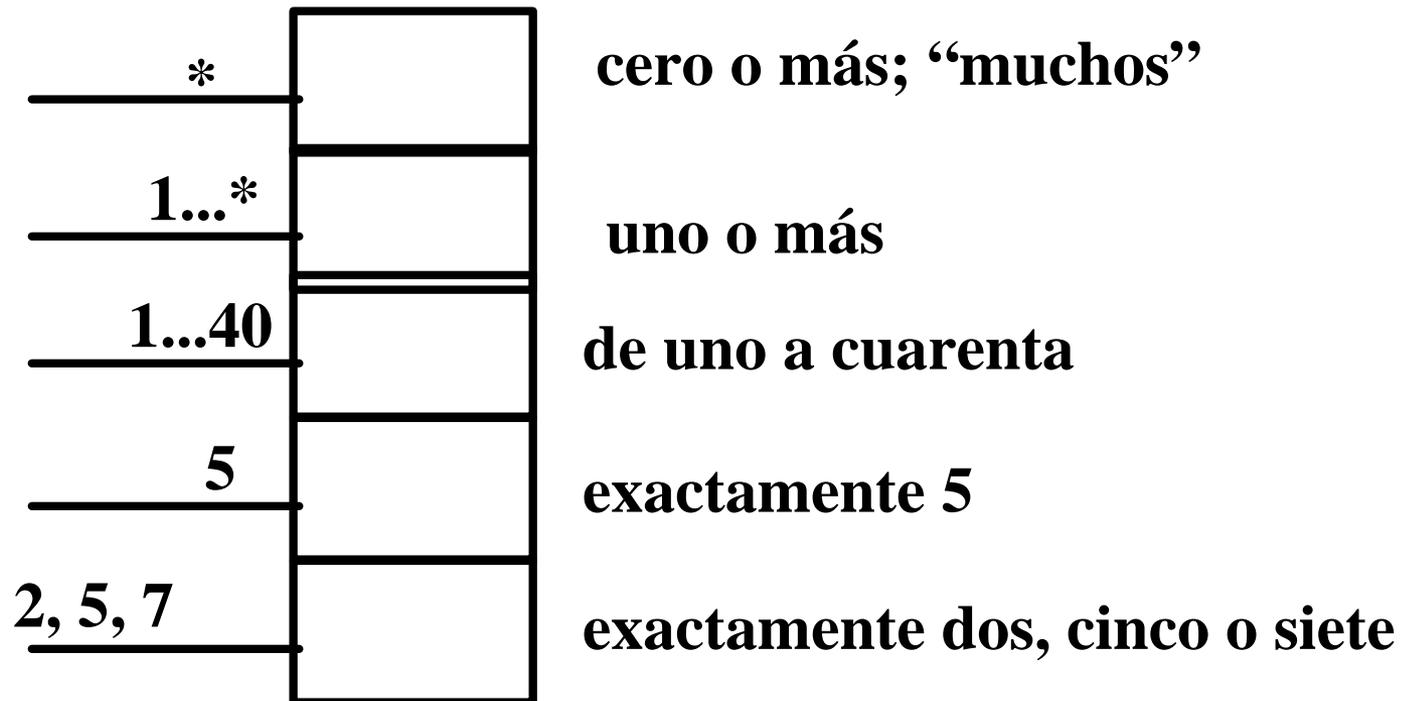


Asociaciones prioritarias

- 1. A es una parte lógica de B (artículo-ley)**
- 2. A es una parte física de B (habitación-casa)**
- 3. A está físicamente contenido en B (producto-estante)**
- 4. A está lógicamente contenido en B (capítulo-ley)**
- 5. A está registrado en B (ladrón-cárcel)**

Multiplicidad

- Define cuantas instancias de una clase pueden asociarse a tantas instancias de otra clase



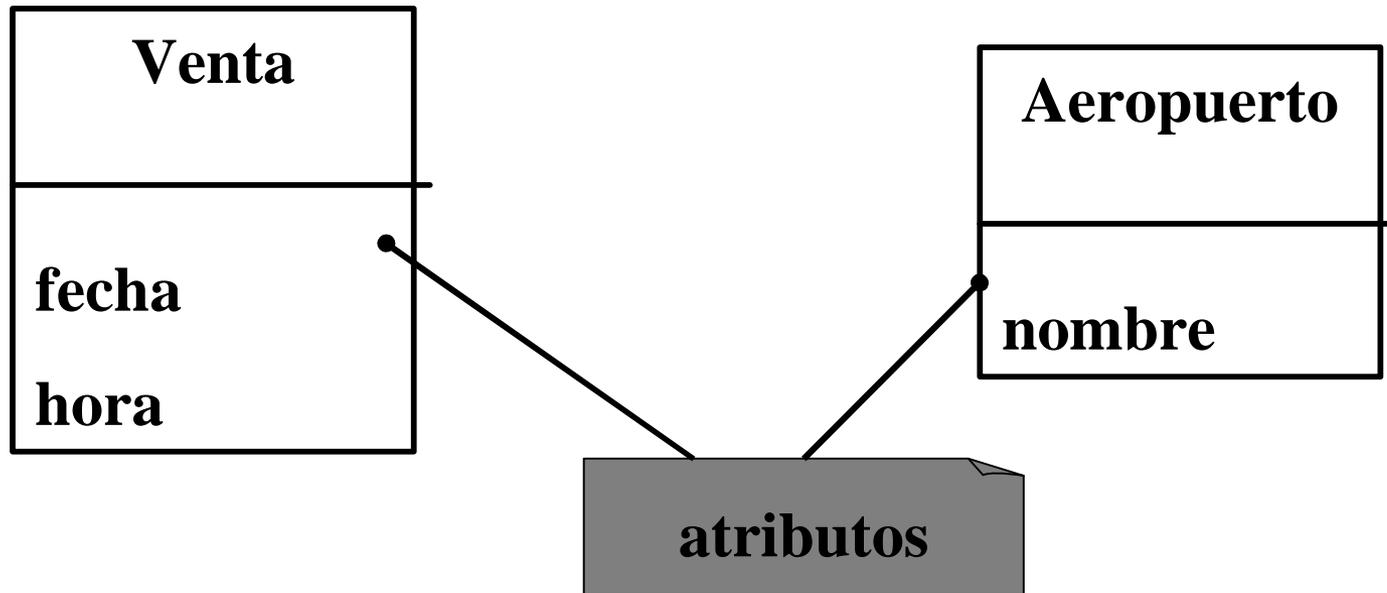
Atributos

- **Es una característica importante de un objeto.**
- **Por ejemplo, un ticket de venta requiere la fecha y la hora.**
- **En consecuencia la clase Venta requiere los atributos fecha y hora**

Atributos comunes

- fecha
- número
- texto
- hora
- booleano
- dirección
- color
- geometría
- número de teléfono
- código de producto
- código postal
- tipos enumerados

Notación de los atributos en UML

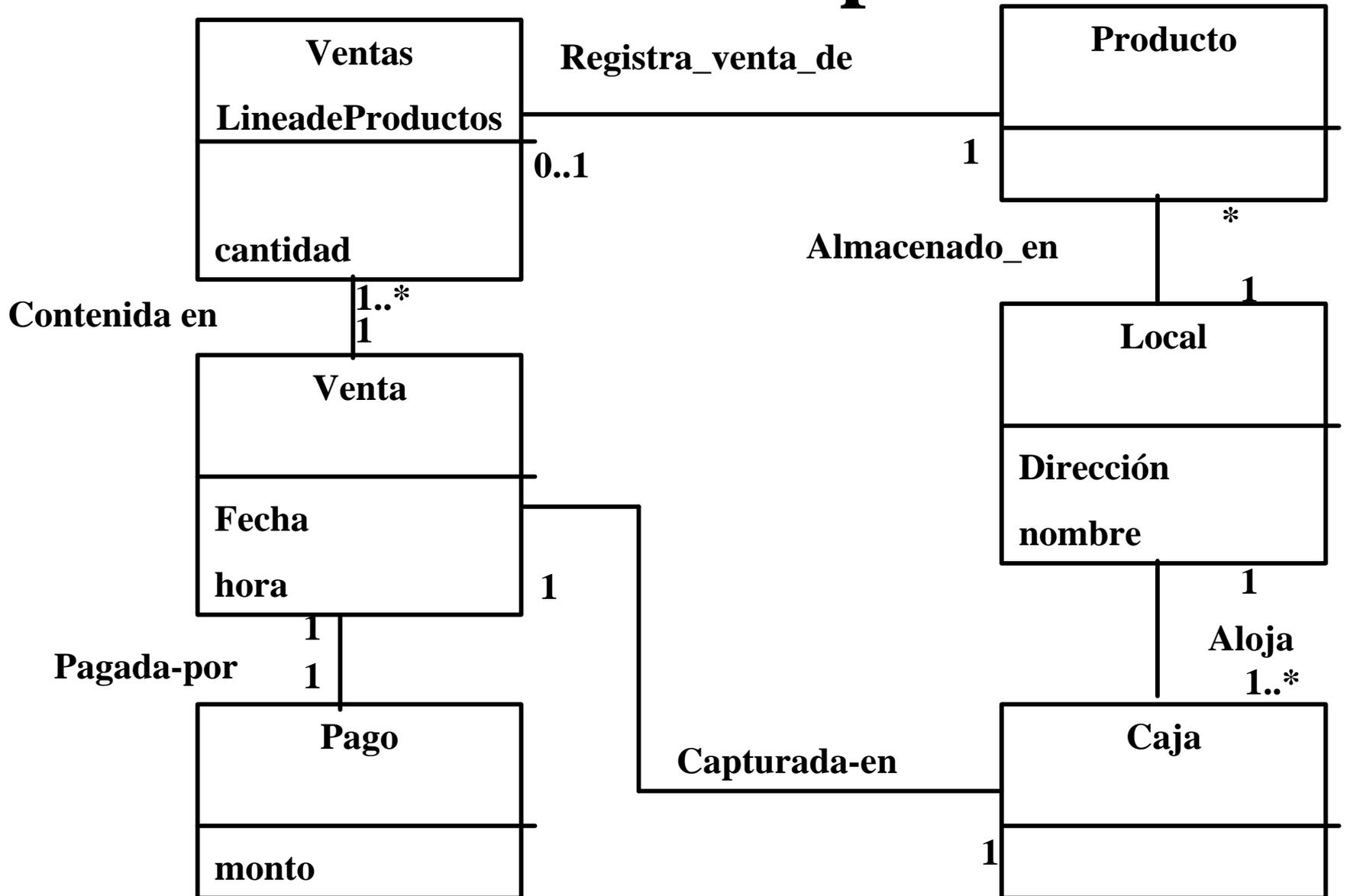


Aplicación

Armemos un Modelo Conceptual

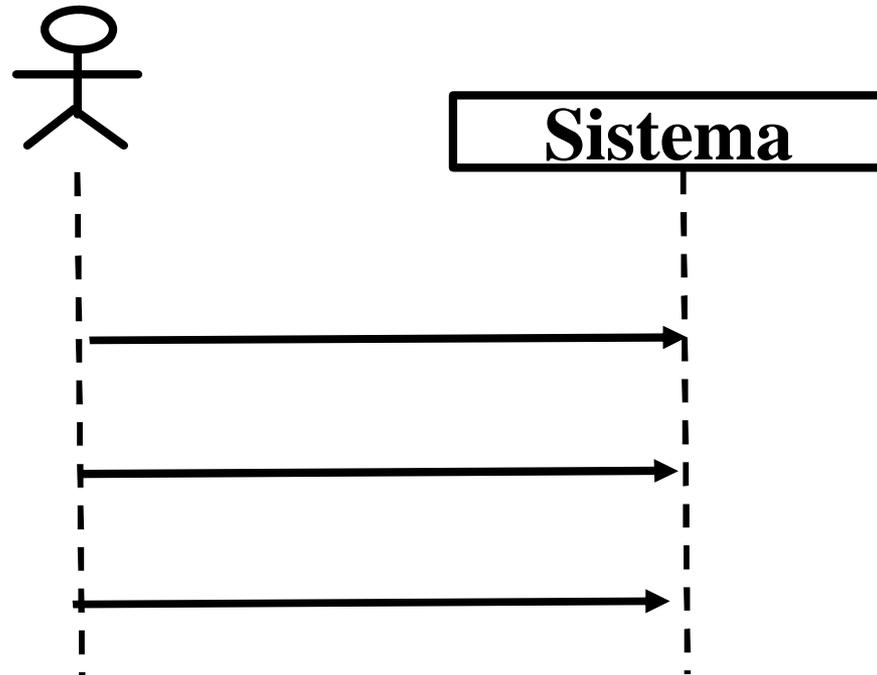
**Tomemos como ejemplo el Video Club
o la Biblioteca**

Modelo conceptual



Comportamiento de los sistemas - Diagramas de secuencia

- Muestran gráficamente los eventos que los actores solicitan al sistema.



Ejemplo

- Se refiere a la secuencia normal de los eventos en el caso típico comprar productos

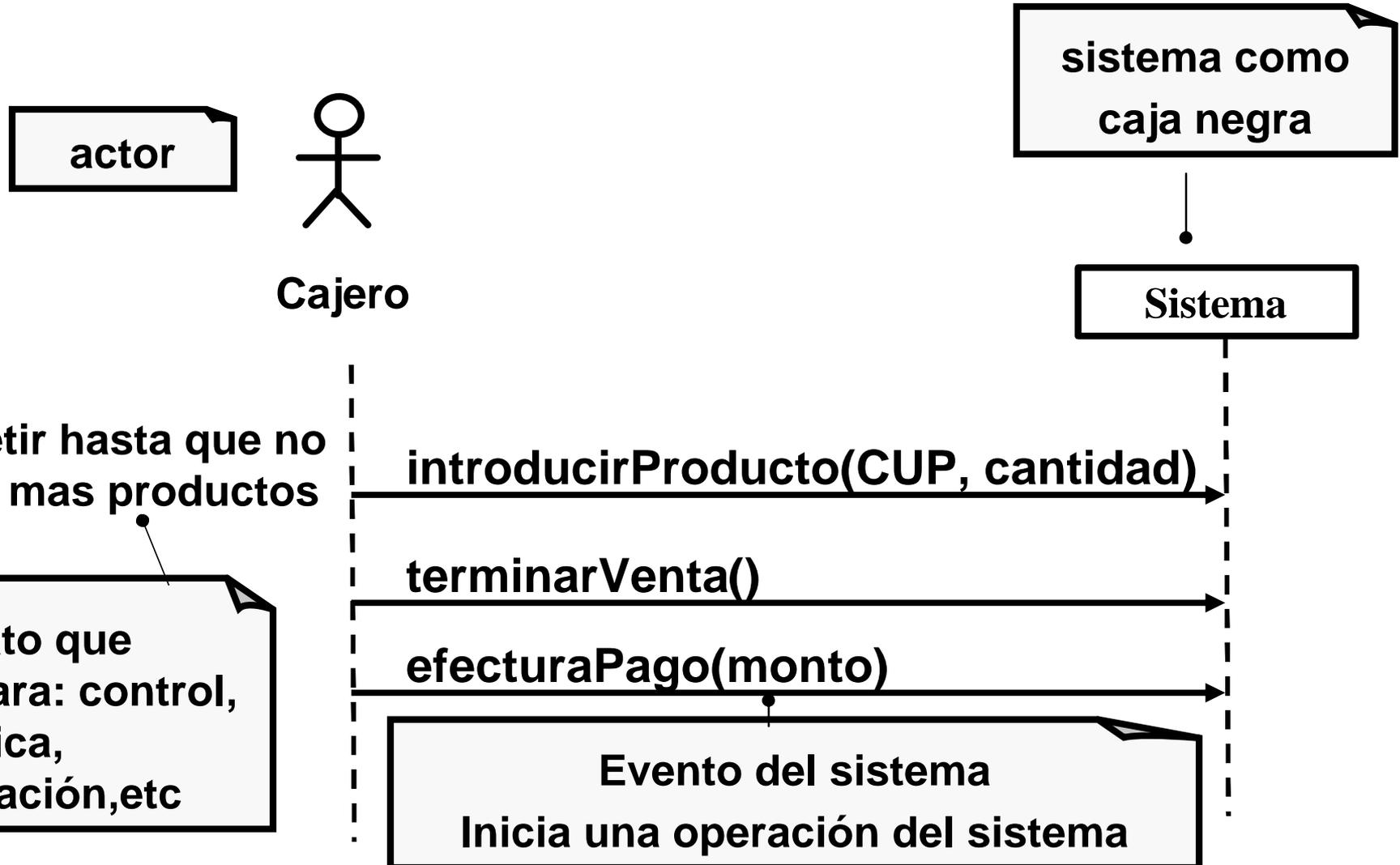


Diagrama de Secuencia Inicial

- **Durante la interacción un actor genera eventos dirigidos a un sistema, solicitando alguna operación a cambio**
- **Su creación depende de la formulación previa de los casos típicos.**
- **Es una descripción de lo que hace, sin explicar cómo lo hace.**
- **Consideramos al sistema como una caja negra.**

Diagrama de Secuencia Inicial

- **Los eventos del sistema pueden incluir parámetros.**
- **Los parámetros son los datos que acompañan la solicitud del actor.**
- **En la aplicación de la caja del supermercado el actor “Cajero” inicia los siguientes eventos:**
 - **introducirProducto**
 - **terminarVenta**
 - **efectuarPago**

Diagrama de Secuencia Inicial

- **El tiempo avanza hacia abajo**
- **El ordenamiento de los eventos debería seguir el orden indicado en el caso típico.**
- **De no ser así deberá reverse el caso típico.**

Eventos y operaciones

- **El evento de un sistema:**
 - Es un hecho externo de entrada que un actor produce en un sistema.
 - Como respuesta se originará una operación del sistema
- **La operación de un sistema**
 - Es una acción que este ejecuta en respuesta a un evento del sistema
- **El nombre del evento y la operación del sistema son idénticos**
 - La diferencia es que uno es el estímulo y el otro la respuesta

Cómo elaborar un diagrama de secuencia

Para describir la secuencia de eventos de un caso típico:

- Trace una línea que represente al sistema como una caja negra
- Identifique a los actores que operan directamente sobre el sistema. Trace una línea para cada uno de ellos
- A partir del caso típico identifique los eventos externos al sistema que son generados por los actores. Muéstrellos gráficamente en el diagrama.
- A la izquierda del diagrama puede incluir o no el texto del caso típico

Asignación de nombres a los eventos

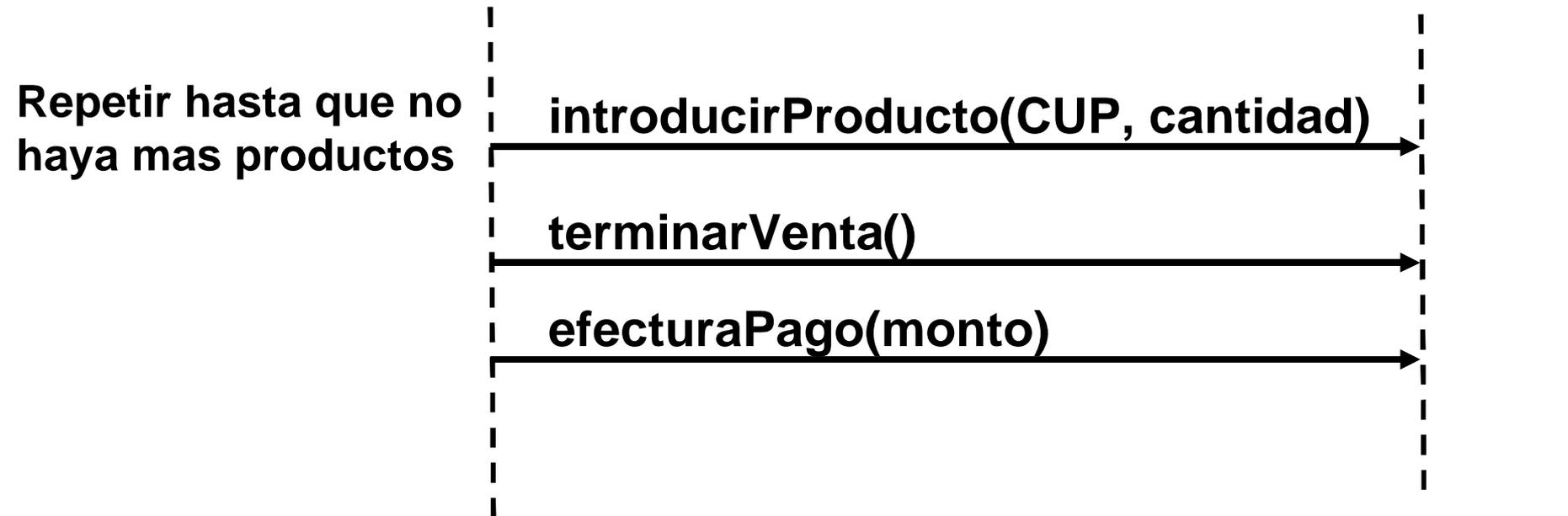
- **Deberían reflejar el propósito**
- **Mejora la claridad si comienza con un verbo (agregar..., introducir..., terminar..., efectuar...)**

Diagrama de secuencia del caso típico comprar productos



Cajero

Sistema



Actividad

- **Confeccionar los diagramas de secuencia para los casos típicos primarios del Video Club o la Biblioteca.**

Contratos

- **Es un documento que describe lo que una operación de sistema se propone hacer.**
- **Se escribe en forma declarativa, *qué* sucederá y *no cómo* se conseguirá.**

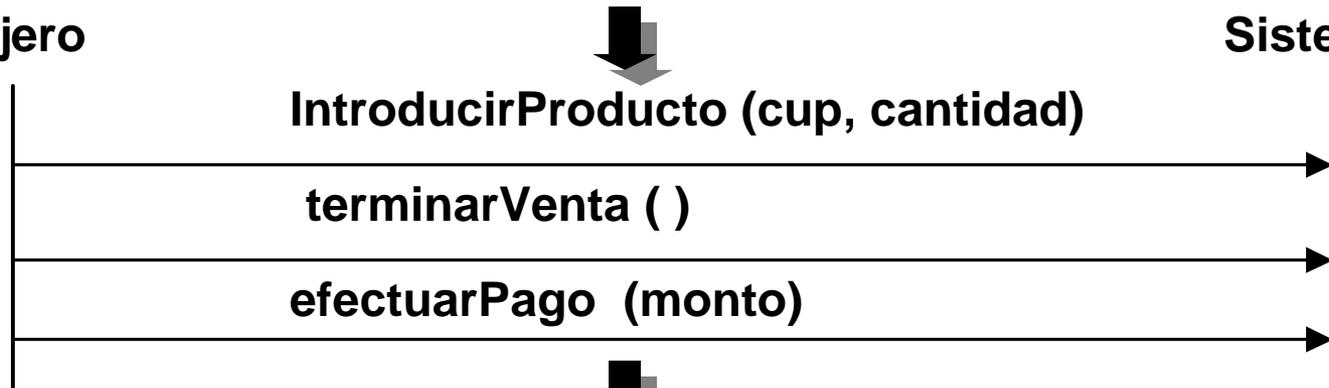
Contratos

Caso Típico: Comprar Productos - Curso Normal de los Eventos

1 Este caso típico comienza...

Cajero

Sistema



Sistema

**terminarVenta()
introducirProducto()
efectuarPago()**

Operación: IntroducirProducto. Se trata de una nueva venta, por lo tanto después de esta operación *fue creada* una Venta...

Secciones del Contrato

Nombre: Nombre de la operación y parámetros.

Responsabilidades: Descripción informal de las responsabilidades que debe cumplir la operación.

Caso: Nombre del Caso Típico

Referencias: N° de referencia de las funciones del sistema, casos típicos, etc.

Notas: notas de diseño, algoritmos, e información afín.

Excepciones: Casos excepcionales.

Secciones del Contrato (cont.)

Salida: Aquello que se espera recibir del sistema (objetivo del contrato).

Precondiciones: Suposiciones acerca del estado del sistema antes de ejecutar la operación.

Poscondiciones: El estado del sistema después de la operación.

Notas

- **Declaraciones de diseño referentes a la operación.**

Ejemplo: la explicación de un algoritmo para manejar la operación (fórmula para calcular la cuota de un préstamo).

Precondiciones

- **Definen la suposición sobre el estado del sistema al iniciarse la operación.**
- **Para describir las precondiciones tener en cuenta lo siguiente:**
 - **Cosas que son importantes probar en el software en algún momento de la ejecución de la operación.**
 - **Cosas de las cuales depende el éxito de la operación.**

Poscondiciones

- **Indican cómo cambió el sistema después de una operación.**
- **Mejora la claridad si se redacta en pretérito (fue creada....).**
- **Describe los cambios necesarios para que el sistema funcione sin necesidad de describir *cómo* se logran.**
- **Nos concentramos en el *qué* debe suceder, no la manera de conseguirlo.**

Poscondiciones

- **Para describir las poscondiciones utilizar las siguientes categorías:**
 - **Creación y eliminación de las instancias.**
 - **Modificación de los atributos.**
 - **Asociaciones formadas y canceladas.**

Contratos para el caso típico comprar productos

Contrato para IntroducirProductos

- **Nombre:** **IntroducirProducto (CUP, cantidad).**
- **Responsabilidades:** **Introducir (registrar) la venta de un producto y agregarlo a la venta. Desplegar la descripción del producto y su precio.**
- **Tipo:** **Sistema.**
- **Referencias cruzadas:****Funciones del sistema R.1.1,R1.3, R1.9**
Caso típico Compara productos.
- **Notas:** **Utilice el acceso super rápido a la base de datos.**
- **Excepciones:** **Si el CUP no es válido, indique que se cometió un error.**
- **Salida:**
- **Precondiciones:** **El sistema conoce el CUP.**

Contratos para el caso típico comprar productos

Contrato para IntroducirProductos (Cont.)

- **Poscondiciones:**
 - Si se trata de una nueva venta, una Venta fue creada (*creación de instancia*).
 - Si se trata de una nueva venta, la nueva Venta fue asociada a la Caja (*asociación formada*).
 - Se creó una instancia VentaLineadeProducto a la Venta (*creación de instancia*).
 - Se asoció VentasLineadeProducto a la Venta (*asociación formada*).
 - Se estableció VentasLineadeProducto.cantidad con el valor cantidad (*modificación de atributo*).
 - La instancia VentasLineadeProducto fue asociada a una EspecificaciondeProducto, basado esto en la correspondencia del CUP (*asociación formada*)

Contratos para el caso típico comprar productos

Contrato para TerminarVenta

- **Nombre:** TerminarVenta
- **Responsabilidades:** Registrar que es el final de la captura de los productos de la venta y desplegar el total de la venta.
- **Tipo:** Sistema.
- **Referencias cruzadas:** Funciones del sistema R.1.2
Caso típico Compra productos.
- **Notas:** Si no se está realizando una venta indicar que se cometió un error.
- **Excepciones:** Si el CUP no es válido, indique que se cometió un error.
- **Salida:**
- **Precondiciones:** El sistema conoce el CUP.

Contratos para el caso típico comprar productos

Contrato para Terminar Venta (Cont.)

- **Poscondiciones:**
 - **Estableció Venta. Esta Terminada como verdadero (modificación de atributo)**

Contratos para el caso típico comprar productos

Contrato para EfectuarPago

- **Nombre:** EfectuarPago (monto)
- **Responsabilidades:** Registrar el pago, calcular el saldo e imprimir el recibo.
- **Tipo:** Sistema.
- **Referencias cruzadas:**Funciones del sistema R.2.1
Caso típico Compra productos.
- **Notas:**
- **Excepciones:** Si la venta no está concluida, indique que se cometió un error.
- **Salida:** Ticket
- **Precondiciones:**

Contratos para el caso típico comprar productos

Contrato para EfectuarPago (Cont.)

- **Poscondiciones:**
 - **Se creó un Pago (*creación de instancia*).**
 - **Se asignó a Pago.MontoOfrecido el valor de monto (*modificación de atributo*).**
 - **Se asoció el Pago a la Venta (*asociación formada*).**
 - **Se asoció la Venta a la Caja para agregarla al registro histórico de las ventas terminadas (*asociación formada*)**

Cómo preparar un contrato

- Identificar las operaciones del sistema a partir de los diagramas de secuencias.
- Elaborar un contrato en cada operación del sistema.
- Comenzar redactando la sección *responsabilidades*, describiendo el propósito de la operación.

Cómo preparar un contrato (cont.)

- Completar la sección *poscondiciones* describiendo los cambios de estado de los objetos en el modelo conceptual.
- Para describir las *poscondiciones* utilizar las siguientes categorías.
 - Creación y eliminación de las instancias.
 - Modificación de los atributos.
 - Asociaciones formadas y canceladas.

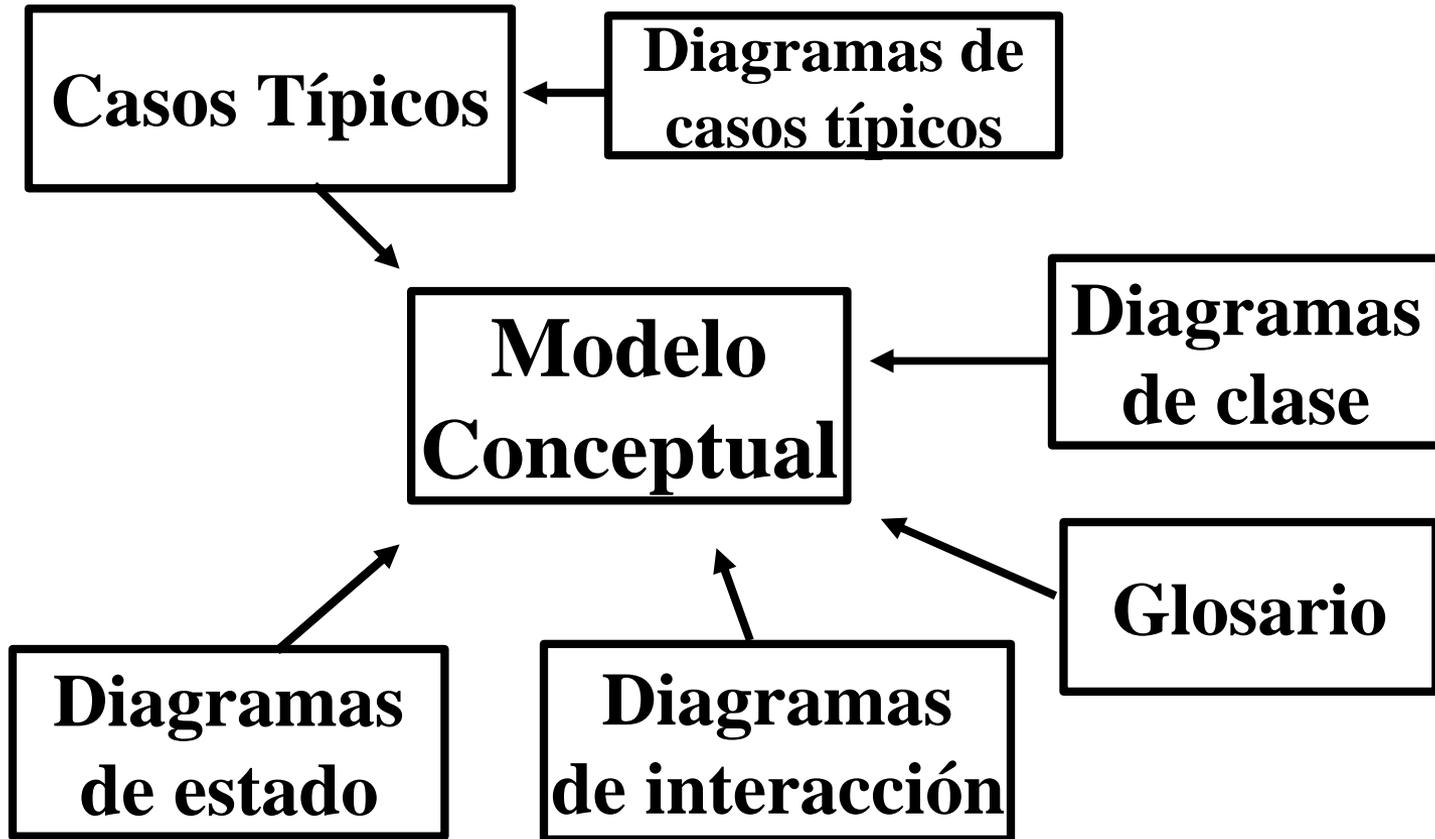
Actividad

- **Confeccionar los principales items de las operaciones del sistema referentes a los diagramas de secuencia del video.**

Conclusión de la fase de análisis

Artefactos del análisis	Preguntas que se contestan
Casos típicos	¿Cuáles son los procesos de la aplicación?
Modelo conceptual	¿Cuáles son los conceptos los términos?
Diagramas de secuencia	¿Cuáles son los eventos y las operaciones del sistema?
Contratos	¿Qué hacen las operaciones del sistema?

Modelo Conceptual



Mensajes y métodos

- **Los objetos no son conjuntos de datos pasivos**
- **Pueden interactuar entre sí**
- **Se comunican a través de mensajes**
- **Cada objeto que recibe un mensaje lo atiende con un método (comportamiento)**
- **El conjunto de mensajes que cada objeto puede atender se denomina interface.**

Actividades del Sistema

Diagramas de Actividad:

- Diagrama de secuencia:

basado en el tiempo

formato en progresión vertical

- Diagrama de colaboración:

basado en el espacio

formato en red

Diagrama de colaboración

- 1. Hacer un diagrama por cada operación**
- 2. Si es muy complejo, subdividir en más simples**
- 3. Empezar desde las responsabilidades**
- 4. Tener en cuenta las postcondiciones**
- 5. Considerar la descripción del caso típico**

Diagrama de colaboración

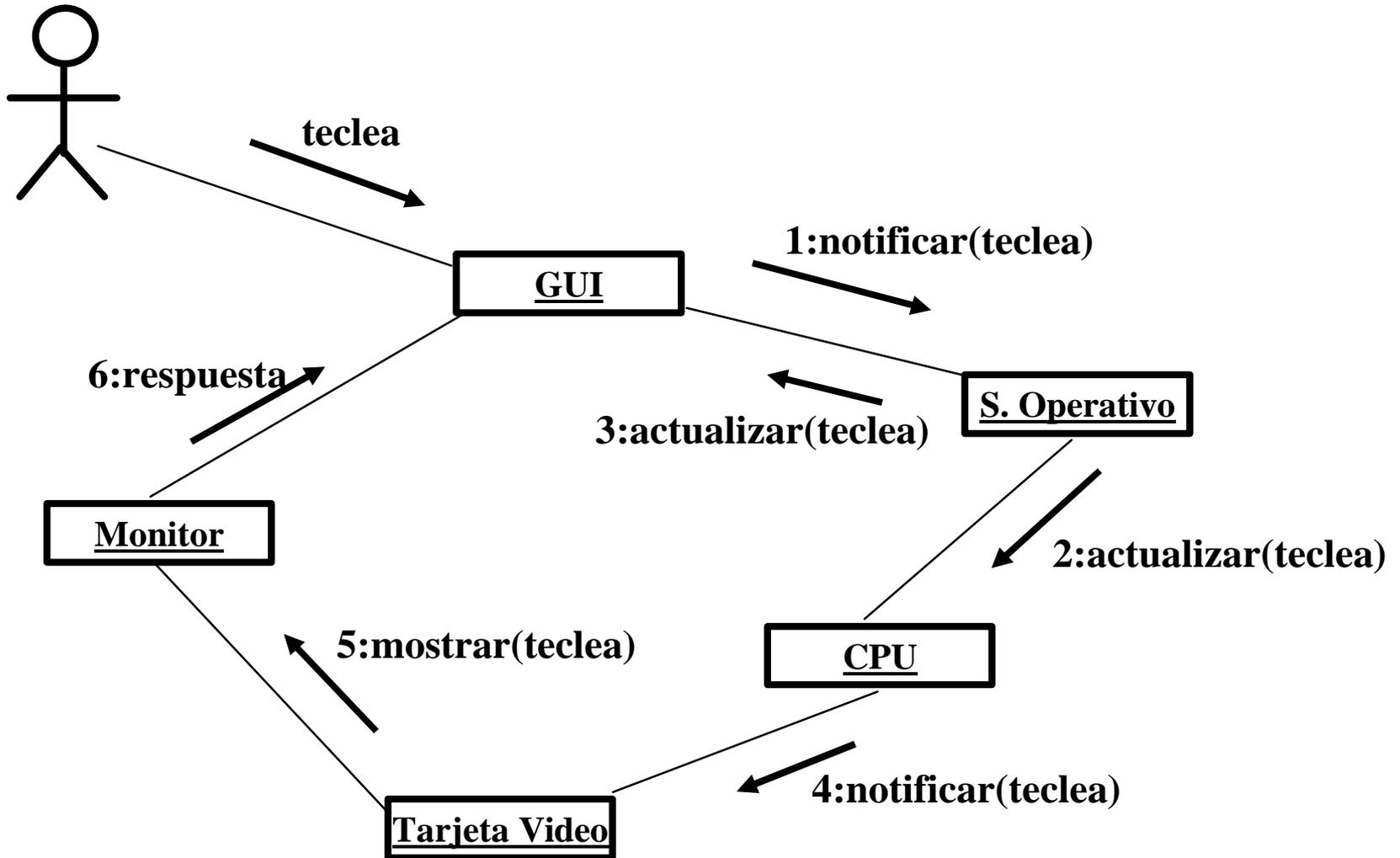
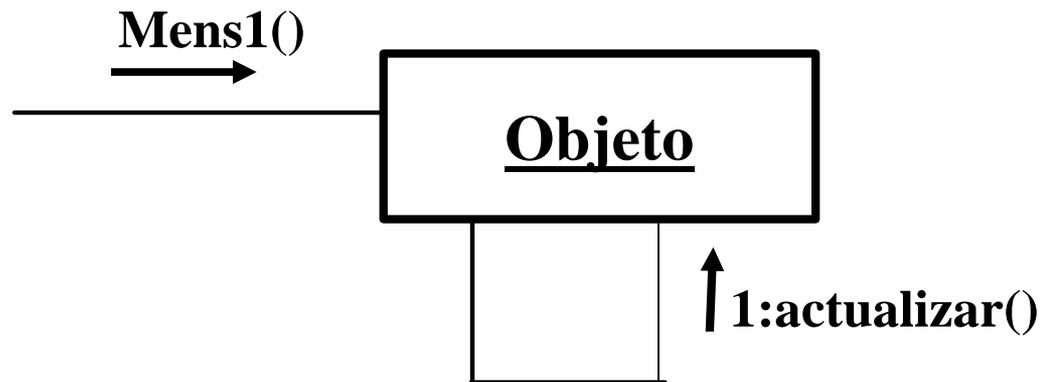


Diagrama de colaboración

Sintaxis de los mensajes:

Retorno : mensaje(parámetro : tipoParámetro) : tipoRetorno

Mensajes a sí mismo (o a “esto”):



Iteración:

Se agrega un asterisco (*) al número de secuencia

Diagrama de colaboración

Mensajes condicionales:

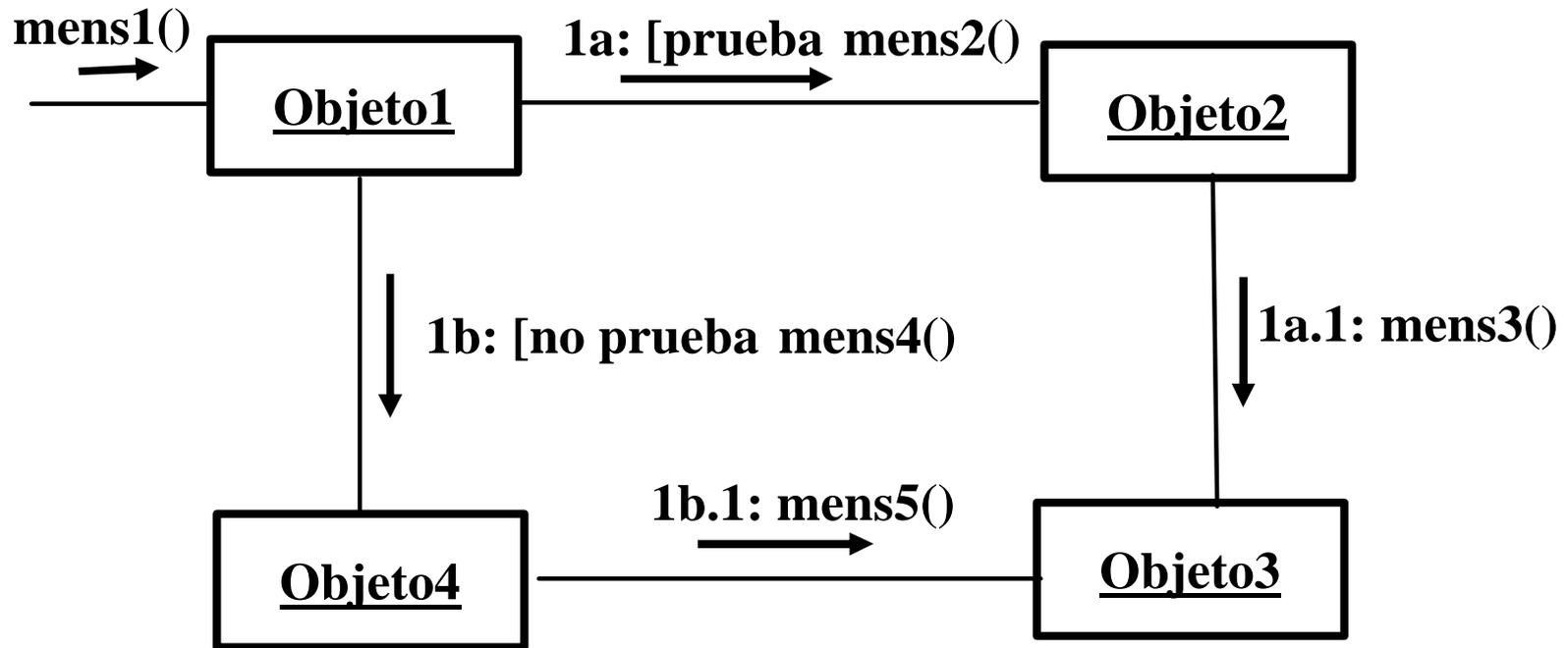


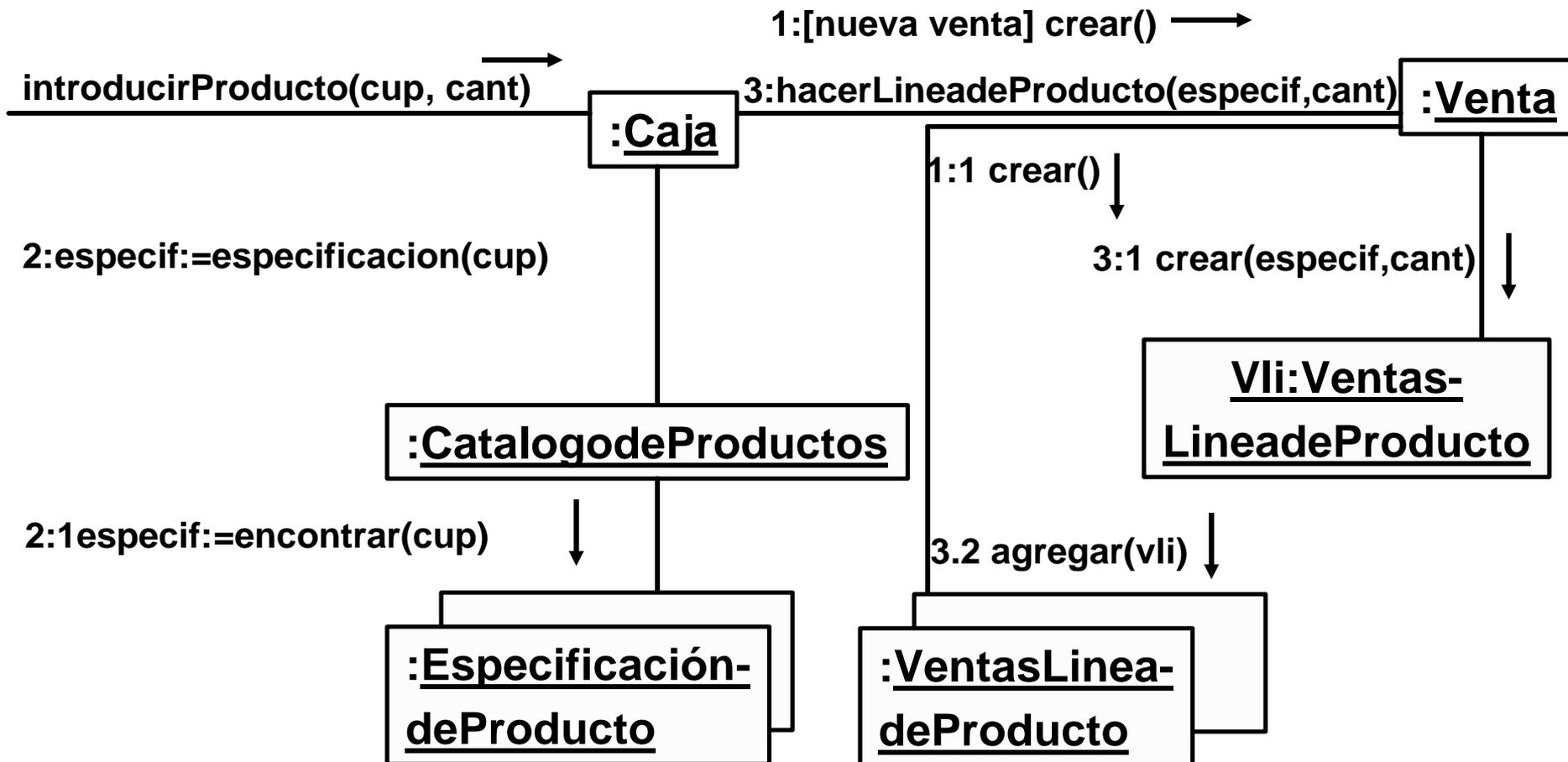
Diagrama de colaboración

Multiobjetos (conjunto de instancias):



El mensaje dirigido a un multiobjeto no se transmite a todos los elementos

Diagrama de colaboración IntroducirProducto



Venta
fecha hora
estaTerminada seTermina() hacerLineadeProducto() efectuarPago() Total()

Caja
terminarVenta() introducirProducto() efectuarPago()

Diagrama de secuencia

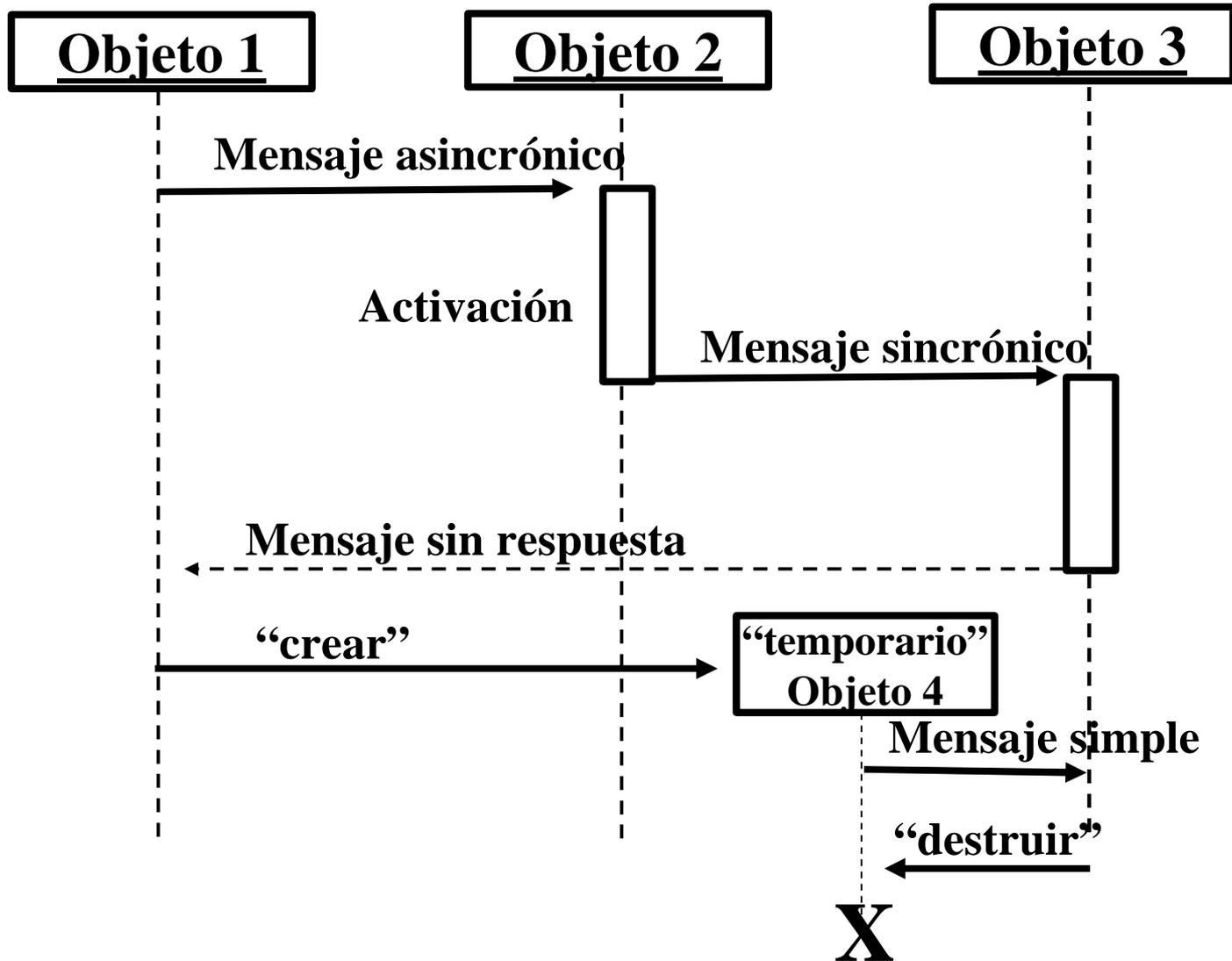


Diagrama de Estado

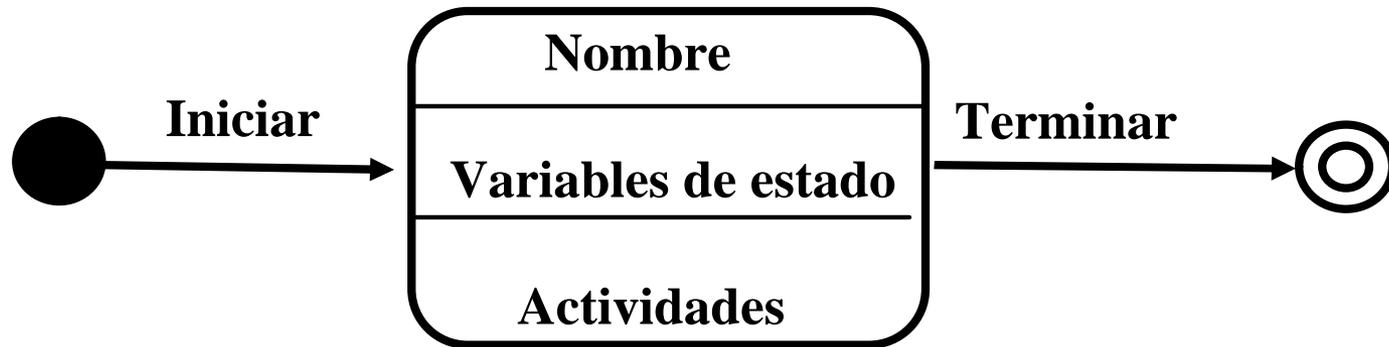
El diagrama de estado del UML describe los eventos y estados más relevantes de un objeto, así como su comportamiento ante cada evento.

Evento: es un acontecimiento u ocurrencia notable, que desencadena un cambio de estado.

Estado: es la condición de un objeto en un momento determinado, o el tiempo que transcurre entre dos eventos.

Transición: es una relación entre dos estados. Cuando ocurre un evento, el objeto pasa de un estado al siguiente.

Diagrama de Estado



- Posibles diagramas** {
- * Casos típicos (procesos)
 - * Sistemas
 - * Ventanas
 - * Coordinadores de aplicaciones
 - * Controladores
 - * Transacciones
 - * Dispositivos
 - * Mutadores

Diagramas de Clases (I)

Cliente

Mozo

Mesa

Cocinero

**Adicio-
nista**

Cocina

Menú

Vajilla

Salón

Diagramas de Clases (II)

