

# Programación Avanzada

## **Análisis**

Especificación del  
Comportamiento del Sistema

# [Contenido]

- n Introducción
- n Modelo de Casos de Uso
- n La Clase Sistema
- n Interacciones con el Sistema
- n Contratos de Software

# Introducción

- n Durante esta actividad de análisis se busca describir en forma precisa cuál debe ser el comportamiento esperado del sistema
- n Se trabaja sobre el Modelo de Casos de Uso
  - i Viendo al sistema como una unidad
  - i Se definen protocolos que caractericen el uso del sistema por parte de los actores en cada escenario de los casos de uso
  - i El comportamiento completo del sistema es especificado al especificar cada mensaje de los protocolos

## [ Introducción (2) ]

- n Cada escenario de los casos de uso a analizar es entendido en términos de una interacción entre los actores involucrados y el sistema
- n Al describir el significado de cada uno de los mensajes identificados en cada interacción se está especificando el comportamiento del sistema

# [ Introducción (3) ]

- n Nos enfocamos en qué es lo que debe hacer el sistema ante cada mensaje
- n La forma en cómo el sistema resuelve internamente un mensaje será definida durante la etapa de diseño

# [Modelo de Casos de Uso]

## n Contenido:

### i Introducción

- n Breve descripción textual que sirve como introducción al modelo

### i Relevamiento de funcionalidades

- n Descripción textual de información no reflejada en el resto del modelo, por ejemplo:
  - i Secuencias típicas en que los casos de uso son utilizados por los usuarios
  - i Otras funcionalidades no capturadas en los casos de uso

# [ Modelo de Casos de Uso (2) ]

- n Contenido (cont.)

- i Actores

- n Todos los actores detectados para el sistema

- i Casos de uso

- n Todos los casos de uso definidos

- i Relaciones

- n Todas las asociaciones entre actores y CU

- i Comportamiento

- n Especificación del comportamiento de cada caso de uso en el modelo, el cual está definido por: Eventos del Sistema y Contratos de Software

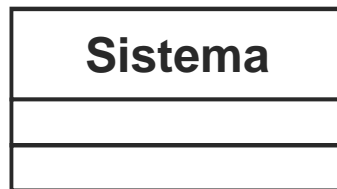
# [ La Clase Sistema ]

- n Durante esta actividad el sistema será considerado como un objeto:
  - i Que es instancia de una clase Sistema
  - i Que tiene operaciones (puede recibir mensajes)
  - i Que tiene un estado
- n En todo Modelo de Casos de Uso se asume que existe una clase Sistema



# [ La Clase Sistema (2) ]

- n Existe una **única instancia** de esta clase la cual representa al “sistema entero”

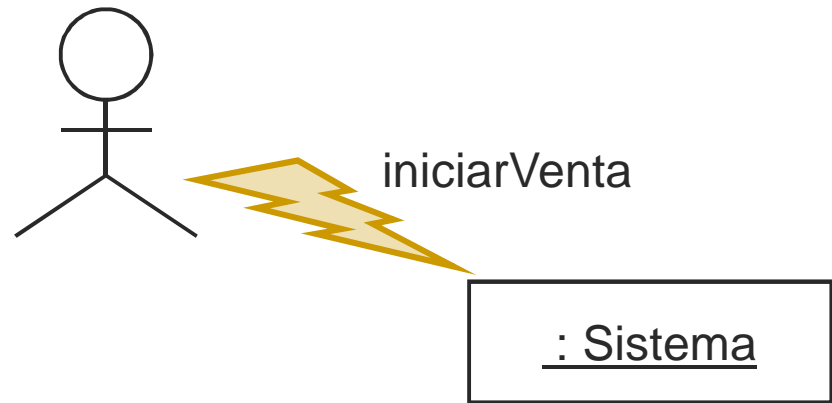
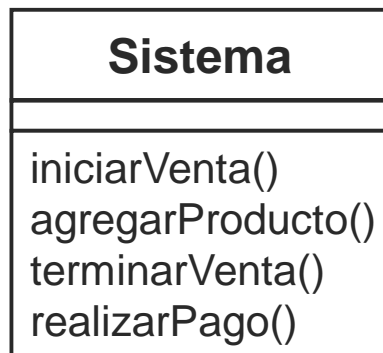


# [ La Clase Sistema (3) ]

- n Las **operaciones** de esta clase permiten que el sistema reciba mensajes de los actores:
  - i Se identifican al definir los protocolos que representan los escenarios de los diferentes casos de uso
  - i Durante el análisis no se busca diseñarlas
  - i Su semántica es definida en términos del efecto que deben tener sobre el estado del sistema

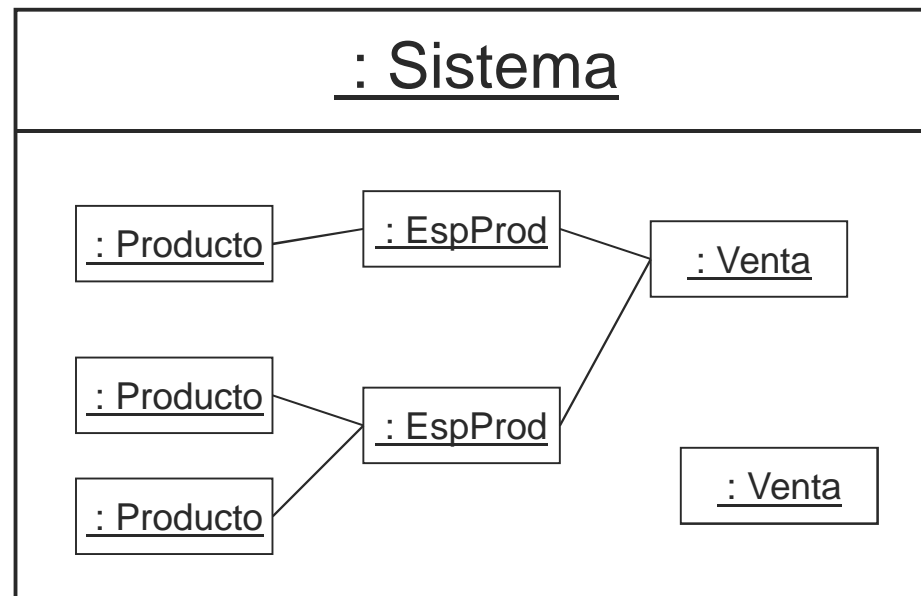
# [ La Clase Sistema (4) ]

- n Un actor puede enviar mensajes al sistema “invocando” sus operaciones



# [ La Clase Sistema (5) ]

- n En esta actividad el **estado** del sistema se asume como una configuración de objetos válida respecto al Modelo de Dominio



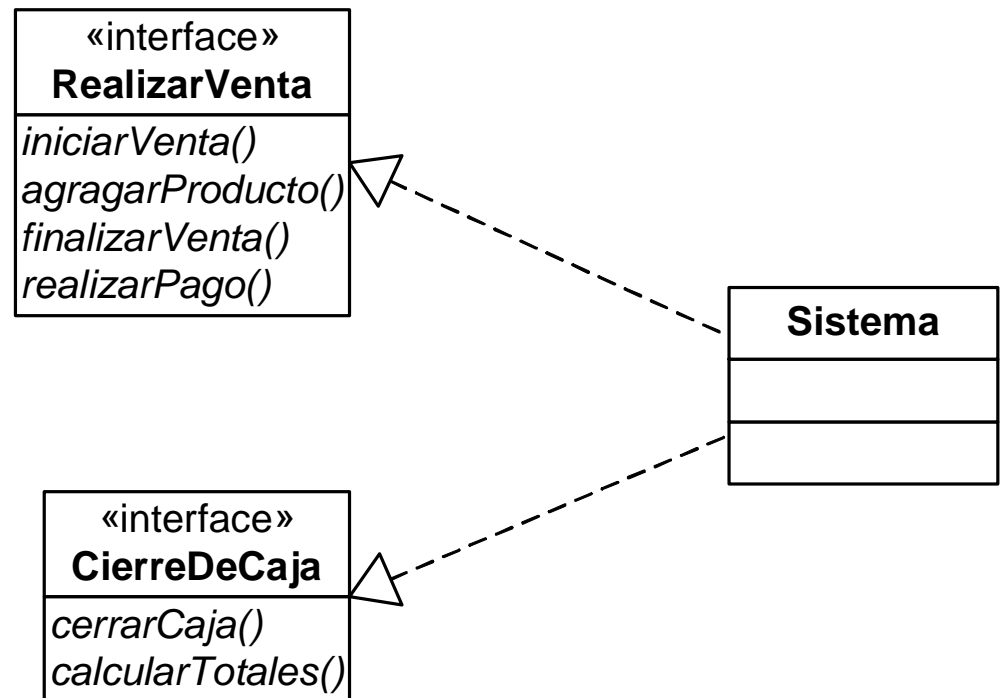
# [ La Clase Sistema (6) ]

- n Dado que no todos los actores participan en todos los casos de uso la visibilidad sobre las operaciones del sistema debe ser limitada
- n Por tanto la clase sistema podría realizar diferentes interfaces
- n Cada interfaz contendría las operaciones utilizadas en un (conjunto de) caso(s) de uso
- n Las operaciones se encontrarían organizadas y los actores verían al sistema a través de la(s) interface(s) que le corresponde(n)

# [ La Clase Sistema (7) ]

El actor **Cajero** usará al sistema solamente a través de esta interfaz

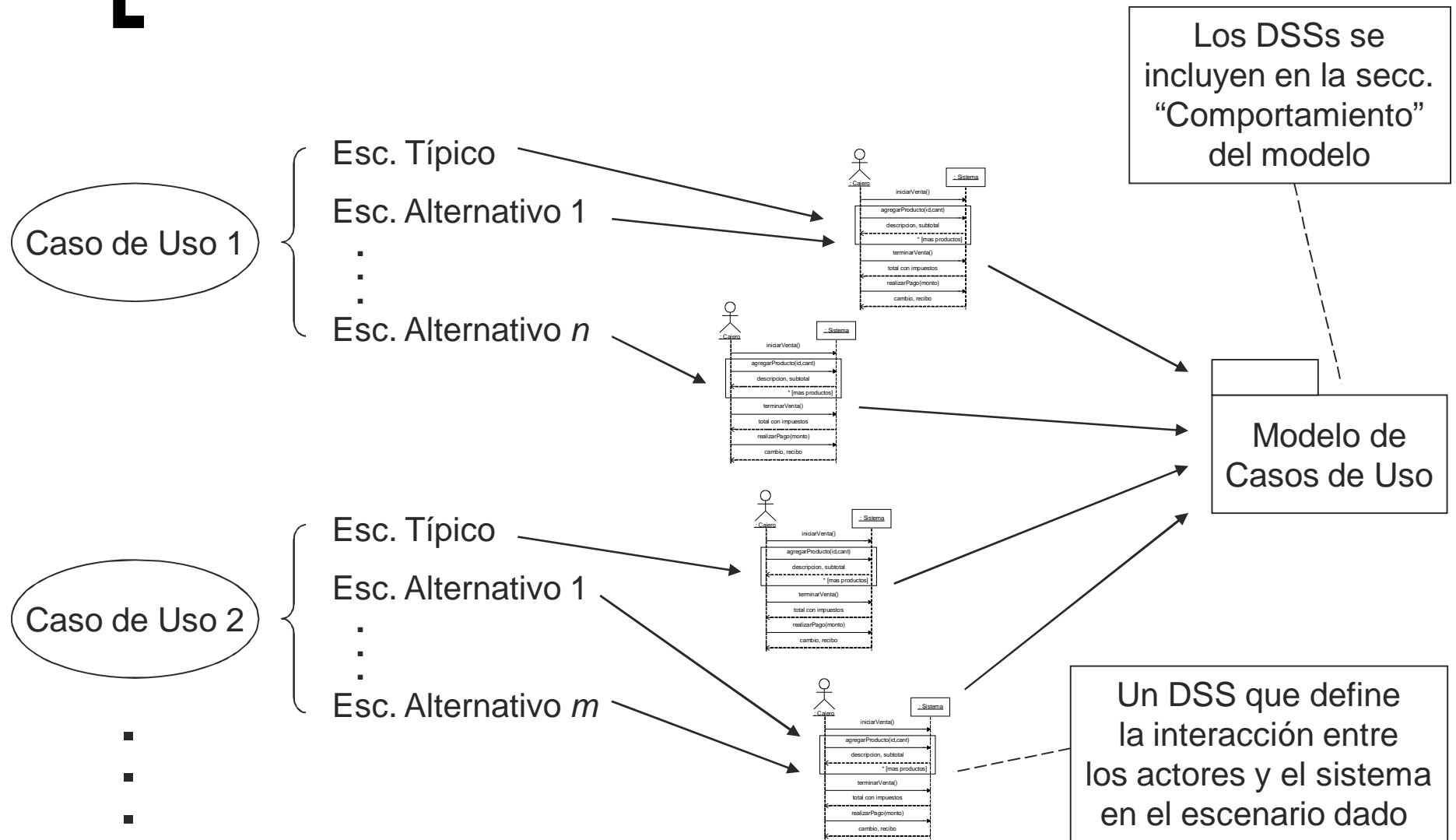
El actor **Supervisor** usará al sistema solamente a través de esta interfaz



# Interacciones con el Sistema

- n Los casos de uso describen la forma en que actores utilizan al sistema para cumplir con sus objetivos
- n Es necesario expresar estas ideas desde un punto de vista técnico
- n Para ello se definen protocolos que determinan la interacción entre los actores y el sistema, ya sea para uno o varios escenarios de un caso de uso
  - i Cada protocolo es expresado mediante un Diagrama de Secuencia del Sistema (DSS)

# Interacciones con el Sistema (2)





# Interacciones con el Sistema

## Eventos del Sistema

- n Un evento del sistema ...
  - i Es un estímulo externo,
  - i Es generado por un actor, y
  - i Ante el cual el sistema debe reaccionar
- n Las acciones de los actores (sobre el sistema) descritas en los casos de uso sugieren los eventos del sistema
- n Es necesario considerar la definición de evento del sistema para identificarlos

# Interacciones con el Sistema

## Eventos del Sistema (2)

### n Ejemplo:

- i *“El Cliente llega a la caja con artículos para comprar”*

Es un evento externo pero no afecta al sistema

**Ø No es un evento del sistema**

- i *“El Cajero ingresa el identificador del producto”*

Es un estímulo externo generado por un actor ante el cual el sistema debe reaccionar

**Ø Es un evento del sistema**

# Interacciones con el Sistema Operaciones del Sistema

- n Los eventos del sistema disparan una operación del sistema
- n Estas operaciones son ejecutadas por la “instancia sistema” en respuesta a la ocurrencia de un evento del sistema
- n Las operaciones del sistema relativas a uno o varios escenarios de un caso de uso permiten definir la interacción entre los actores y el sistema

# Interacciones con el Sistema

## Operaciones del Sistema (2)

- n Las operaciones del sistema pueden tener asociados parámetros
- n Ejemplo:
  - i “*El Cajero ingresa el identificador del producto*”  
representa un evento que dispara la operación  
`void agregarProducto(ident: String)`
  - i “*El Cajero ... hasta terminar los productos*”  
representa un evento que dispara la operación  
`void terminarVenta()`

# Interacciones con el Sistema

## Diag. de Secuencia del Sistema

- n Es un artefacto incluido en el Modelo de Casos de Uso que define e ilustra la interacción entre los actores y el sistema en uno o varios escenarios de un CU
- n Incluye:
  - i Una instancia representando a cada participante (sistema y actores)
  - i Los mensajes enviados entre ellos en el/los escenario/s correspondiente/s (con sus respuestas)

## [ Interacciones con el Sistema Diag. de Secuencia del Sistema (2) ]

- n Un Diagrama de Secuencia del Sistema puede ser construido para:
  - i Un escenario de un Caso de Uso
  - i Varios escenarios de un Caso de Uso
- n Un criterio para decidir entre estas alternativas será la complejidad de estos escenarios y la simplicidad (o no) del DSS resultante

# Interacciones con el Sistema

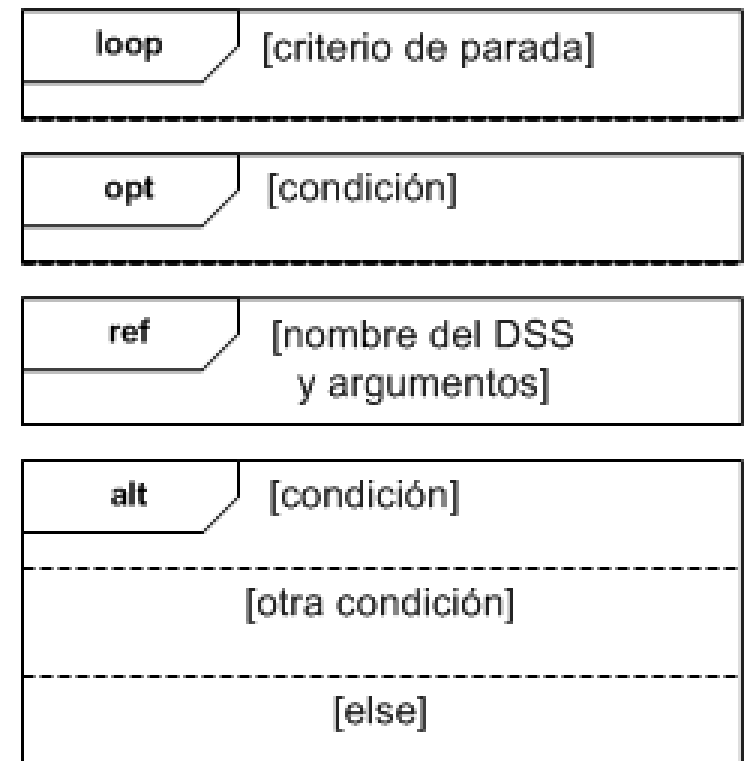
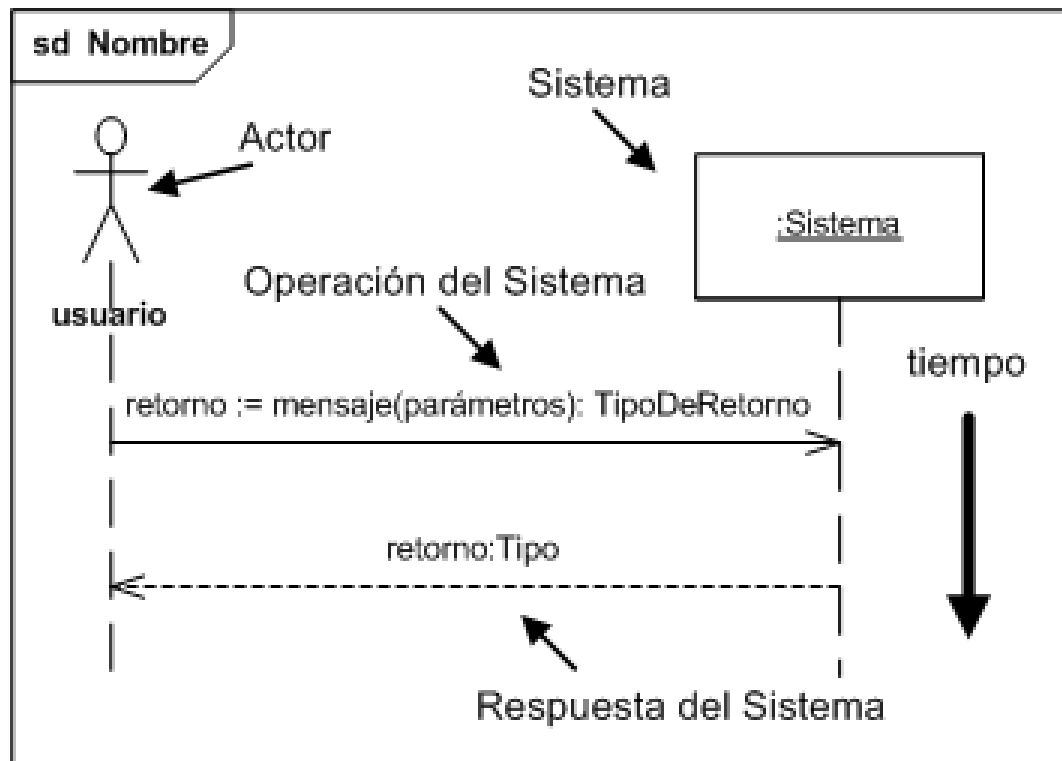
## Diag. de Secuencia del Sistema (3)

- n Los diagramas de secuencia del sistema definen la conversación entre los actores y el sistema, enfocándose en los mensajes que el sistema recibe
- n Sería posible incluir además mensajes enviados desde el sistema hacia los actores:
  - i Sin embargo esto no forma parte del conjunto de servicios que el sistema brinda (y cuya especificación es el objetivo de la presente actividad)

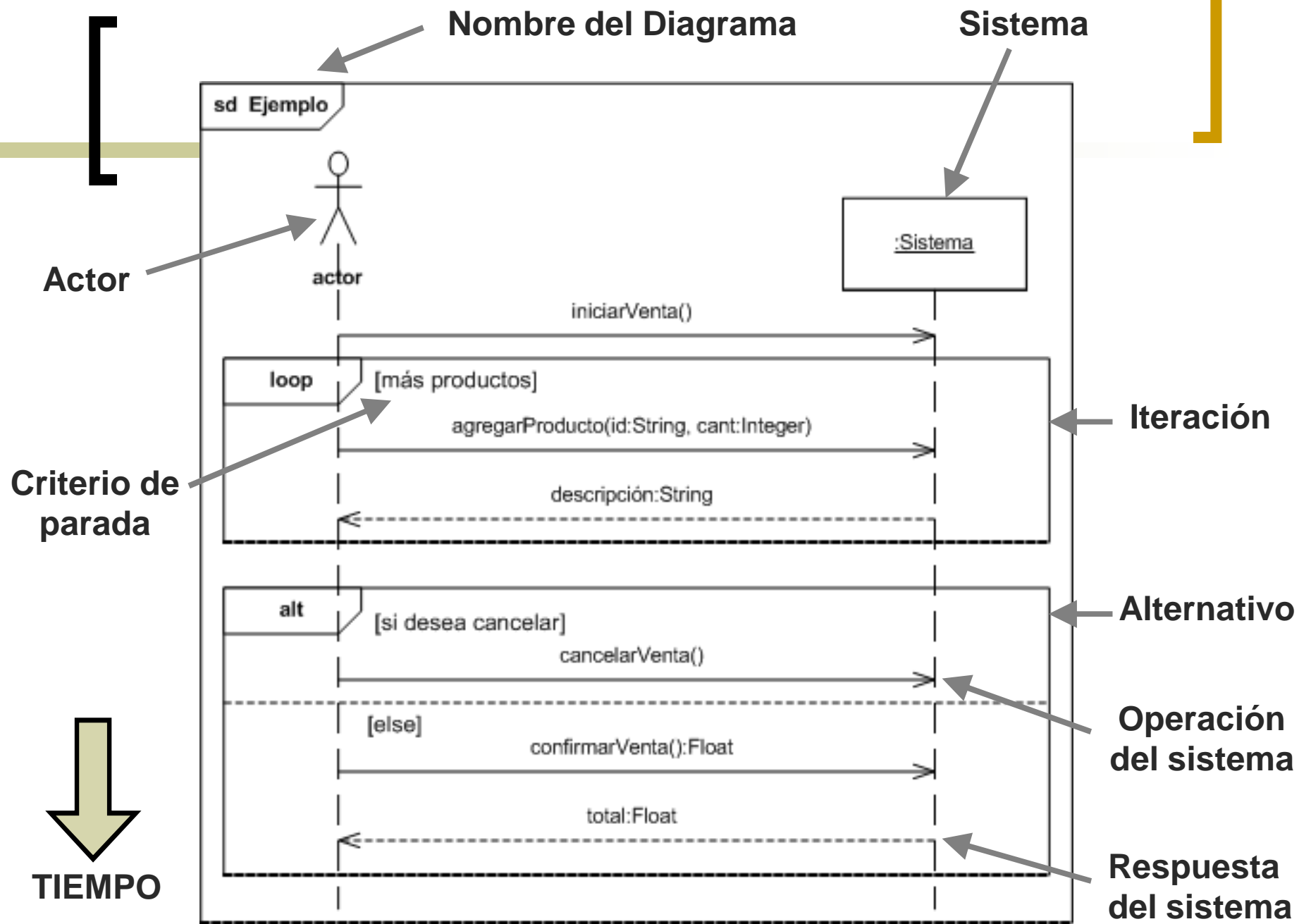
# Interacciones con el Sistema

## Diag. de Secuencia del Sistema (4)

### n Notación:







# Interacciones con el Sistema

## Sugerencias

### n Definición de un DSS:

1. Incluir una instancia que represente al sistema como una unidad
2. Identificar cada actor que participe en el/los escenario/s considerado/s e incluir una instancia para cada uno
3. De la descripción del caso de uso identificar aquellos eventos que los actores generen y sean de interés para el sistema e incluir cada uno de ellos como un mensaje

# Interacciones con el Sistema

## Sugerencias (2)

- n Límite del sistema:
  - i Para identificar eventos del sistema es útil pensar en el límite del sistema
  - i El límite suele determinarse para que coincida con el sistema de software (y el de hardware también)
  - i Buscar aquello que ocurra fuera de ese límite y que además lo atraviese

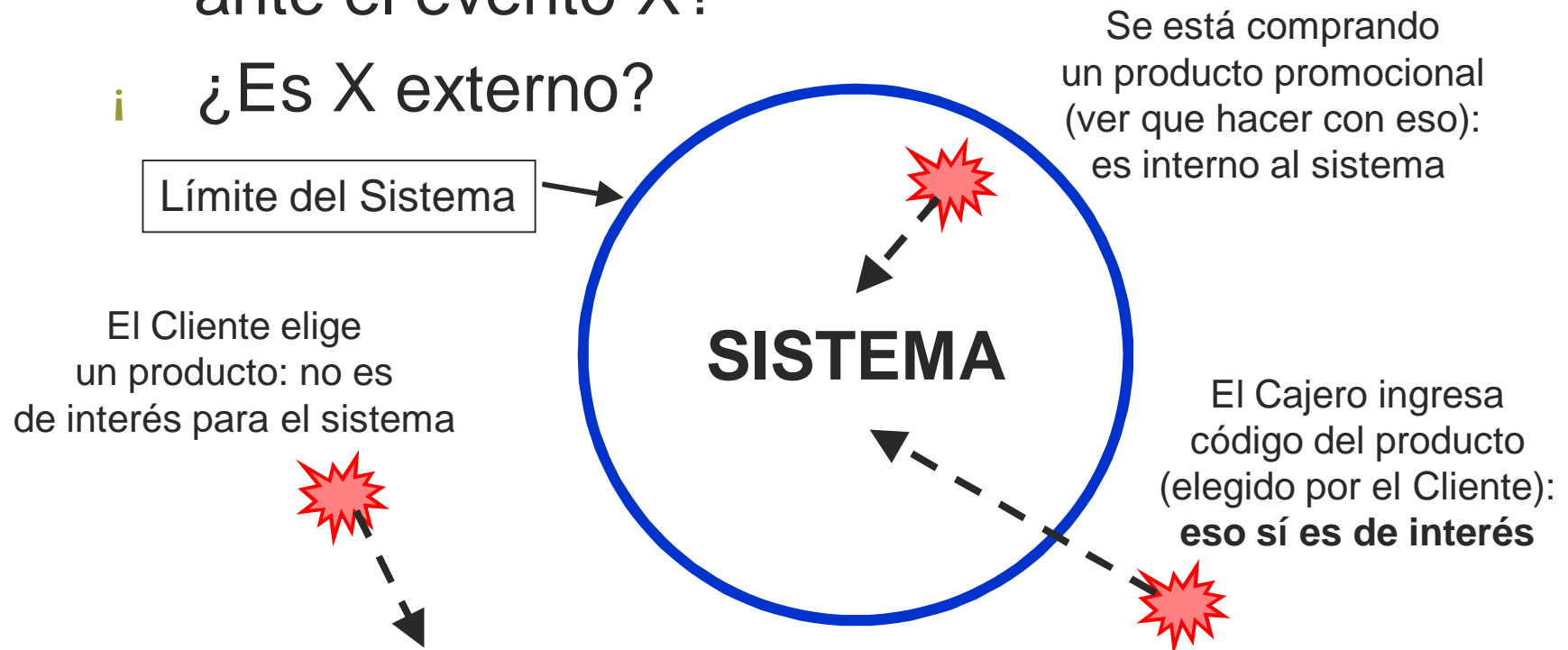
# Interacciones con el Sistema

## Sugerencias (3)

### n Límite del sistema (cont.):

i ¿Es responsabilidad del sistema reaccionar ante el evento X?

i ¿Es X externo?



# Interacciones con el Sistema

## Sugerencias (4)

- n Memoria del Sistema:
  - i El sistema puede (o no) tener memoria:
    - n Sin memoria, los mensajes son independientes
    - n Con memoria, cada mensaje puede “recordar” la información utilizada en un estado previo del sistema
  - i Debe indicarse claramente si el sistema tiene o no memoria, y en caso de tenerla, qué información recuerda

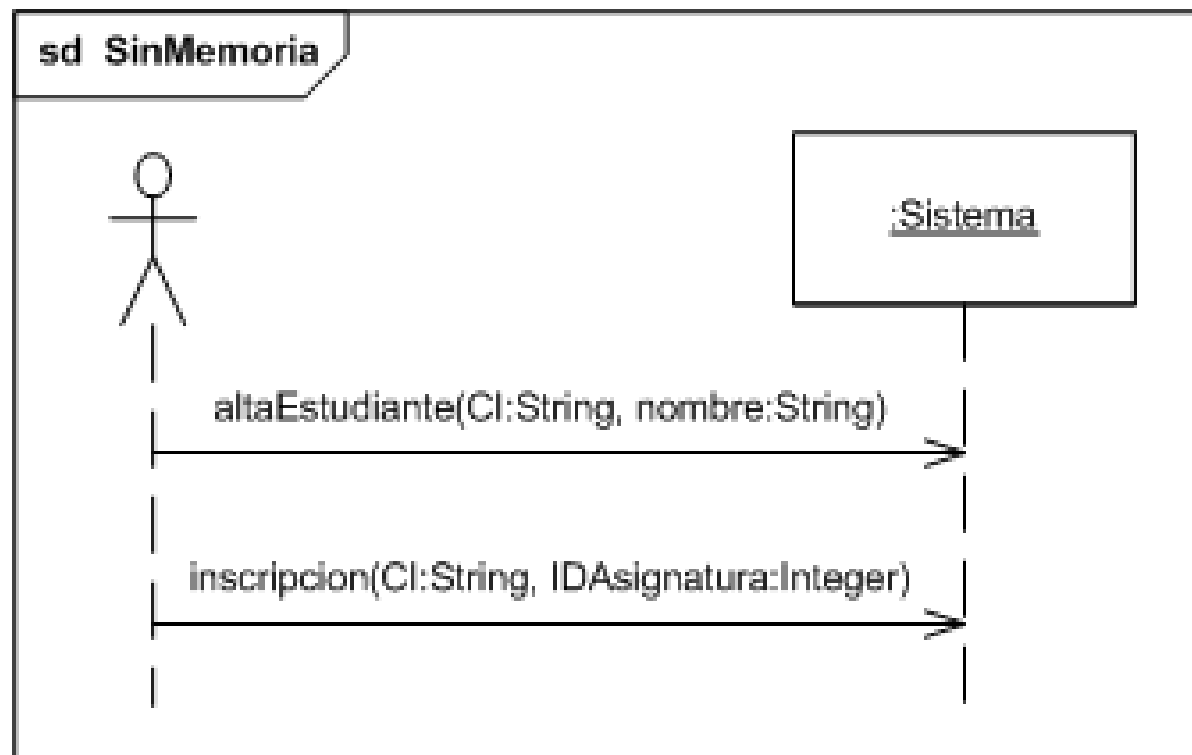
# Interacciones con el Sistema

## Sugerencias (5)

- n Memoria del Sistema (cont.):
  - i Para indicar la memoria de un sistema, generalmente basta con indicarlo en el nombre del diagrama y mediante la utilización de notas en el diagrama
  - i Alternativamente, puede utilizarse un diagrama de estructura estática en aquellos casos en que interese indicar una estructura compleja de dicha memoria

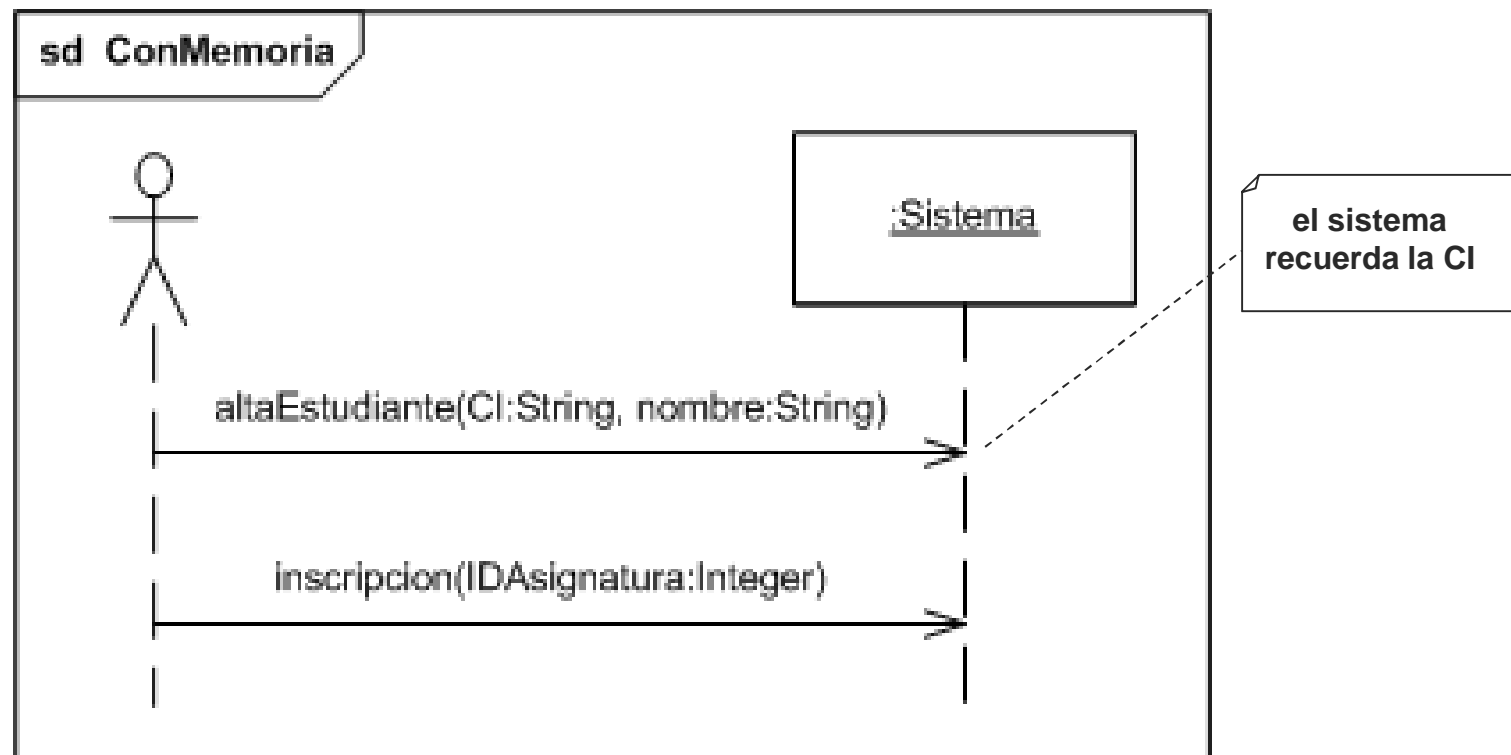
# Interacciones con el Sistema Sugerencias (6)

## n Ejemplo: DSS sin memoria



# Interacciones con el Sistema Sugerencias (7)

## n Ejemplo: DSS con memoria





# Interacciones con el Sistema

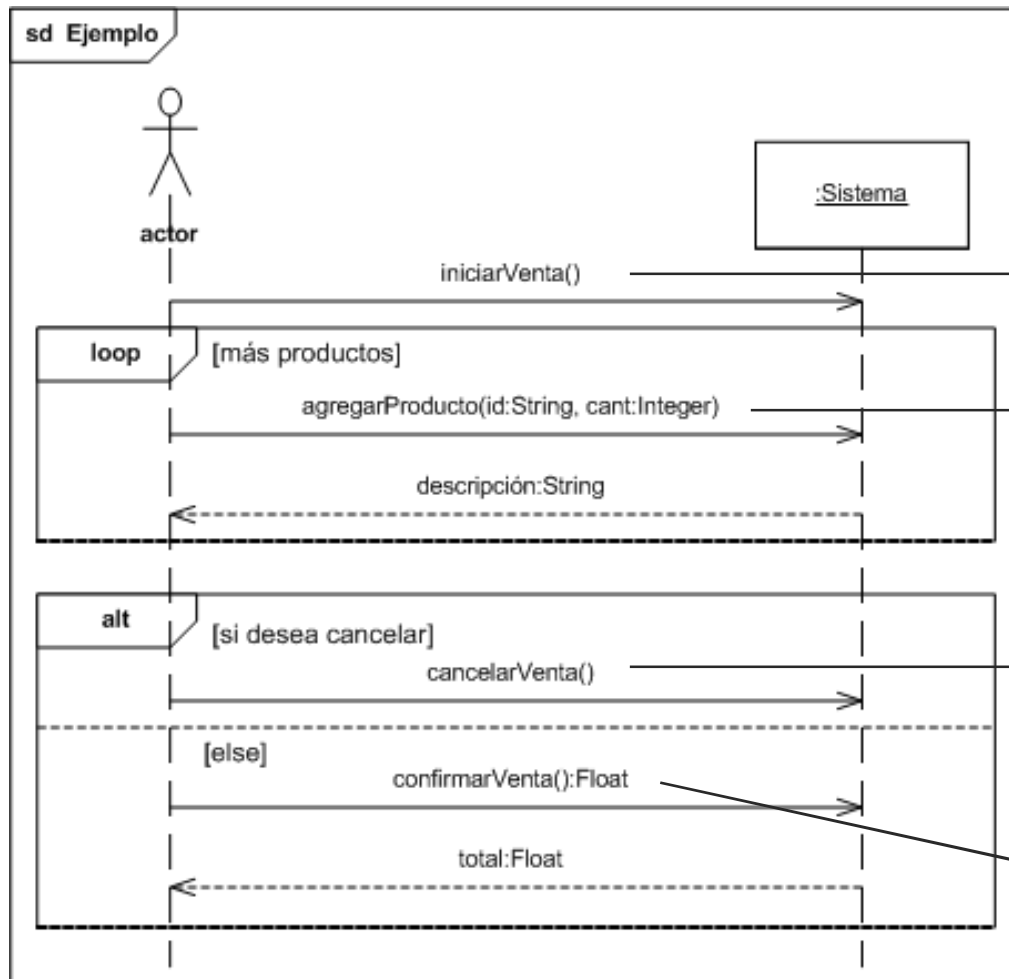
## Errores Comunes

- n Envío de mensajes hacia el usuario
- n Desconocer la memoria del sistema
- n No especificar *data types* utilizados
- n Sobrecargar de información un diagrama de secuencia pudiendo realizar varios de ellos
- n No indicar tipo de parámetros ni valor de retorno de los mensajes

# [ ¿Qué Sigue? ]

- n Una vez identificadas las operaciones del sistema es posible especificar su comportamiento
- n Esta especificación expresa el efecto que una operación tendrá sobre el sistema
- n Para ello se realizará un Contrato de Software para cada operación del sistema

# [ ¿Qué Sigue? (2) ]



Los contratos se incluyen en la secc. "Comportamiento" del modelo

**Contrato:**  
iniciarVenta()

**Contrato:**  
agregarProducto()

**Contrato:**  
cancelarVenta()

**Contrato:**  
confirmarVenta()

Modelo de  
Casos de Uso

# [ Contratos de Software ]

- n Un contrato de software especifica el comportamiento o efecto de una operación
- n La especificación es declarativa y no imperativa
- n Esta técnica está basada en las ternas de Hoare en las que:
  - i Se describen propiedades del resultado, en lugar de dar un conjunto de pasos o instrucciones que indiquen cómo calcularlo

# Contratos de Software

## Enfoque de Contratos

- n El contrato de una operación es un contrato entre partes
  - i **Consumidor** de la operación: quién la invoca
  - i **Proveedor** de la operación: quién la implementa
- n Determina derechos y obligaciones para cada una de las partes

# Contratos de Software

## Enfoque de Contratos (2)

	Obligaciones	Derechos
<b>Consumidor</b>	Satisfacer precondición	Obtener la postcondición satisfecha
<b>Proveedor</b>	Satisfacer postcondición	Procesamiento más simple al poder asumir como satisfecha la precondición

# Contratos de Software

## Enfoque de Contratos (3)

- n El Consumidor se compromete a satisfacer la precondition al invocar la operación:
  - i Si la satisface: tiene derecho a exigir que la postcondición se satisfaga
  - i Si no la satisface: no se le garantiza la correctitud del resultado de la invocación
- n Por esta razón es responsabilidad del Consumidor saber cuándo invocar a la operación (y manejar en forma adecuada el resultado)

# Contratos de Software

## Enfoque de Contratos (4)

- n El Proveedor se compromete a satisfacer la postcondición al finalizar la operación solamente cuando la precondición fue satisfecha al momento de la invocación
- n El compromiso no comprende el caso en que la precondición no fue satisfecha:
  - i En ese caso el Proveedor puede devolver un valor arbitrario y el Consumidor tiene que aceptarlo y saber qué hacer con él



# Contratos de Software

## Enfoque de Contratos (5)

- n Ejemplo “Autorización de Documento”:
  - i Precondición: el documento está en la oficina antes de la hora 10
  - i Postcondición: el documento está firmado por el Gerente a la hora 18
- n Consumidor:
  - i *“Yo te traigo el documento a las 10, pero a las 18 lo quiero firmado”*
- n Proveedor:
  - i *“Yo te hago firmar el documento para las 18, pero lo necesito antes de las 10”*

# Contratos de Software

## Enfoque de Contratos (6)

### n Ejemplo (cont.)

#### i Caso 1 (ambos cumplen)

n El documento llegó a las 9:45

n A las 18 estaba firmado

#### i Caso 2 (el consumidor no cumple)

n El documento llegó a las 11:20

n A las 18 no estaba firmado

#### i Caso 3 (el consumidor cumple pero el proveedor no)

n El documento llegó a las 9:10

n A las 18 no estaba firmado

El proveedor no  
tiene que cumplir

Esto denota un bug  
en la implementación  
del proveedor

# Contratos de Software

## Enfoque de Contratos (7)

### n Consumidor:

- i Prefiere precondiciones débiles: implica menos trabajo
- i Prefiere postcondiciones fuertes: implica más resultados

### n Proveedor:

- i Prefiere precondiciones fuertes: implica menos preocupaciones
- i Prefiere postcondiciones débiles: implica menos trabajo

# Contratos de Software

## Enfoque de Contratos (8)

### n Precondición:

- i Es a lo que debe acceder el Consumidor para obtener el resultado deseado
- i Es lo que debe exigir el Proveedor para llegar al resultado

### n Postcondición:

- i Es a lo que accederá el Consumidor
- i Es a lo que se compromete el Proveedor

# Contratos de Software

## Enfoque de Contratos (9)

- n Tanto las Pre- como las Post- las determina el Proveedor
- n El Consumidor:
  - i Viendo la Post- sabe qué va a obtener (sin saber cómo)
  - i Viendo la Pre- sabe a cambio de qué obtiene el resultado

# Contratos de Software

## Contratos de Operaciones

- n Los contratos se pueden realizar para operaciones de cualquier tipo de clase
- n En esta actividad las realizaremos para operaciones del sistema
- n Para una operación  $X$  tendremos  $\{P\}S\{Q\}$ 
  - i  $P$  es la precondition de  $X$  (especificada)
  - i  $S$  es el programa que implementa  $X$  (a ser diseñado más adelante en la etapa de Diseño)
  - i  $Q$  es la postcondition de  $X$  (especificada)

# Contratos de Software

## Contratos de Operaciones (2)

- n ¿Quién utiliza el contrato (partes P y Q) de una operación?
  - i Un diseñador de nuestro equipo que deba diseñar S
    - n Para saber qué es lo que tiene que lograr su diseño de la operación
    - n En función de lo anterior para decidir cómo será el diseño de la operación (parte S)
  - i Un desarrollador de otro equipo que deba invocar la operación (el diseño o implementación de S no es su responsabilidad)
    - n Para saber qué es lo que la operación hace sin tener que ver el diseño o la implementación de S

# Contratos de Software

## Condiciones

- n ¿En qué términos se expresan las pre- y postcondiciones?  
¿Y para el caso particular de operaciones del sistema?
- n En términos generales estas condiciones refieren al estado del sistema antes y después de la invocación a la operación
  - i Las precondiciones refieren además a los argumentos de la operación
  - i Las postcondiciones refieren además al valor retornado por la operación (si existe)



# Contratos de Software

## Condiciones (2)

- n Las Precondiciones refieren al momento previo a la invocación y expresan condiciones sobre
  - i Los valores de los parámetros de la operación
  - i El estado del sistema:
    - n La creación de objetos
    - n La destrucción de objetos
    - n La conexión de objetos
    - n La desconexión de objetos
    - n La modificación del valor de atributos de objetos

# Contratos de Software

## Condiciones (3)

- n Las Postcondiciones refieren al momento posterior a la invocación expresan condiciones sobre
  - i El valor de retorno (si corresponde)
  - i El estado del sistema:
    - n La creación de objetos
    - n La destrucción de objetos
    - n La conexión de objetos
    - n La desconexión de objetos
    - n La modificación del valor de atributos de objetos

# Contratos de Software

## Condiciones (4)

### n Creación de objetos:

- i Pre: Declarar que el objeto no existe
- i Post: Declarar que el objeto existe

### n Destrucción de objetos:

- i Pre: Declarar que el objeto existe
- i Post: Declarar que el objeto no existe y que todos los objetos que estaban conectados a él ya no lo están

# Contratos de Software

## Condiciones (5)

### n **Conexión de objetos:**

- i Pre: Declarar que los objetos no están conectados
- i Post: Declarar que los objetos están conectados

### n **Desconexión de objetos:**

- i Pre: Declarar que los objetos están conectados
- i Post: Declarar que los objetos no están conectados

### n **Modificación del valor de atributos de objetos:**

- i Pre: Declarar que el objeto exista
- i Post: Declarar que el atributo del objeto tiene el valor dado

# Contratos de Software

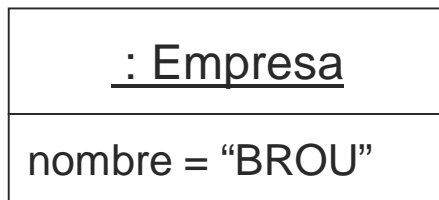
## Condiciones (6)

- n Ejemplo para operación `contratar( )`
  - i **Precondición:** No existe un objeto de tipo Empleado con el valor '6150' en el atributo 'codigo', existe un objeto de tipo Empresa con valor 'BROU' en el atributo 'nombre'
  - i **Postcondición:** Existe un nuevo objeto de tipo Empleado con el valor '6150' en el atributo 'codigo' que está conectado a uno de tipo Empresa que tiene el valor 'BROU' en el atributo 'nombre'
  - i De esto se puede derivar que la operación crea al objeto de tipo Empleado y lo conecta con el de tipo Empresa

# Contratos de Software

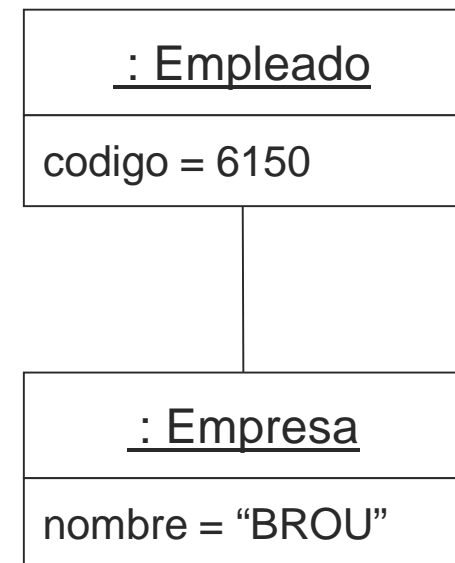
## Condiciones (7)

$\sigma_1$



$\sigma$

2



Se pasa del estado  $\sigma_1$  al estado  $\sigma_2$  mediante la ejecución de  
`contratar("BROU", 6150);`

# Contratos de Software

## Condiciones (8)

- n Notar que el contrato NO dice cómo debe implementarse la operación del sistema `contratar()`
- n Expresa condiciones sobre el estado inicial y sobre el estado final que indican qué es lo que la operación hace, pero no cómo lo hace

# Contratos de Software

## Estructura de Contratos

- n Un contrato es un artefacto textual que se incluye en la sección 'Comportamiento' del Modelo de Casos de Uso
- n Está estructurado de la siguiente forma:
  - i **Firma:** Cabezal sintáctico de la operación
  - i **Parámetros:** Descripción de los parámetros de la operación
  - i **Responsabilidades:** Descripción de las responsabilidades, una idea de lo que debe realizar la operación



# Contratos de Software

## Estructura de Contratos (2)

### n Estructura (cont.)

- i **Referencias cruzadas:** Caso(s) de Uso a los que pertenece la operación
- i **Salida:** Resultado de la operación (sólo si es una función)
- i **Precondición:** Descripción del estado de la instancia del sistema a la que se le aplicará la operación, y otras condiciones que sea necesario asumir previo a la aplicación (por ejemplo, con respecto a los parámetros)

# Contratos de Software

## Estructura de Contratos (3)

### n Estructura (cont.)

- i **Postcondición:** Descripción del estado de la instancia del sistema a la que se le aplicó la operación
- i **Snapshots:** (Opcional)
  - n Pares de *snapshots* que ejemplifiquen el estado de la instancia a la que se le aplicó la invocación, previo y posterior a la invocación
  - n La invocación concreta que produce el cambio ejemplificado (mostrando los parámetros efectivos)

# Contratos de Software

## Errores Comunes

- n Incluir invariantes como postcondiciones
- n Omitir el resultado de una operación como postcondición

# [ ¿Qué Sigue? ]

- n Hasta el momento se tienen identificadas y especificadas las operaciones del sistema para todos los casos de uso definidos
- n Es posible ahora realizar un diseño en el que
  - i Se identifiquen los objetos que realmente participarán en la solución
  - i Se definan interacciones entre dichos objetos tal que cada una cumpla un contrato correspondiente a una operación del sistema