



Ingeniería de Software II (CI-4712)

UML (Unified Modeling Language)

Octubre de 2007

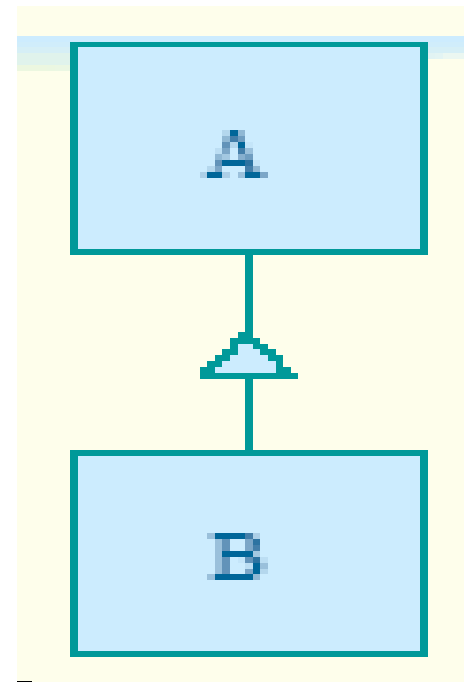
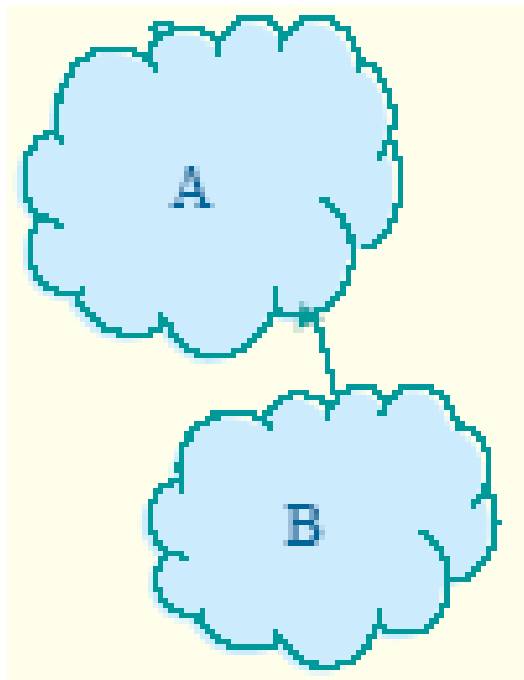
UML

un modelo o pieza de información
producido en el proceso de
desarrollo de software

- Un lenguaje para especificar, visualizar y construir *artefactos* de sistemas de software
- **No** es una metodología.

Antecedentes

- Proliferación de métodos OO
- Diferencias de notaciones



Puntos de convergencia

- Coincidencias
- Unificación de conceptos y notaciones
- Búsqueda de estándares



Lenguaje de Modelación

Evolución

UML 2.0

UML 1.4

UML 1.3

UML 1.1

UML 1.0

UML 0.9 & 0.91

Método Unificado 0.8

Booch'93

OMT-2

Booch'91

OMT-1

OOSE

Publicación de UML 1.1
Septiembre 1997

Publicación de UML 1.0
Enero 1997

Junio 96 y Octubre 1996

Estandarización

Unificación

Colaboradores y
expertos

Fragmentación

Documentos públicos

septiembre de 2001

Abril 1999:

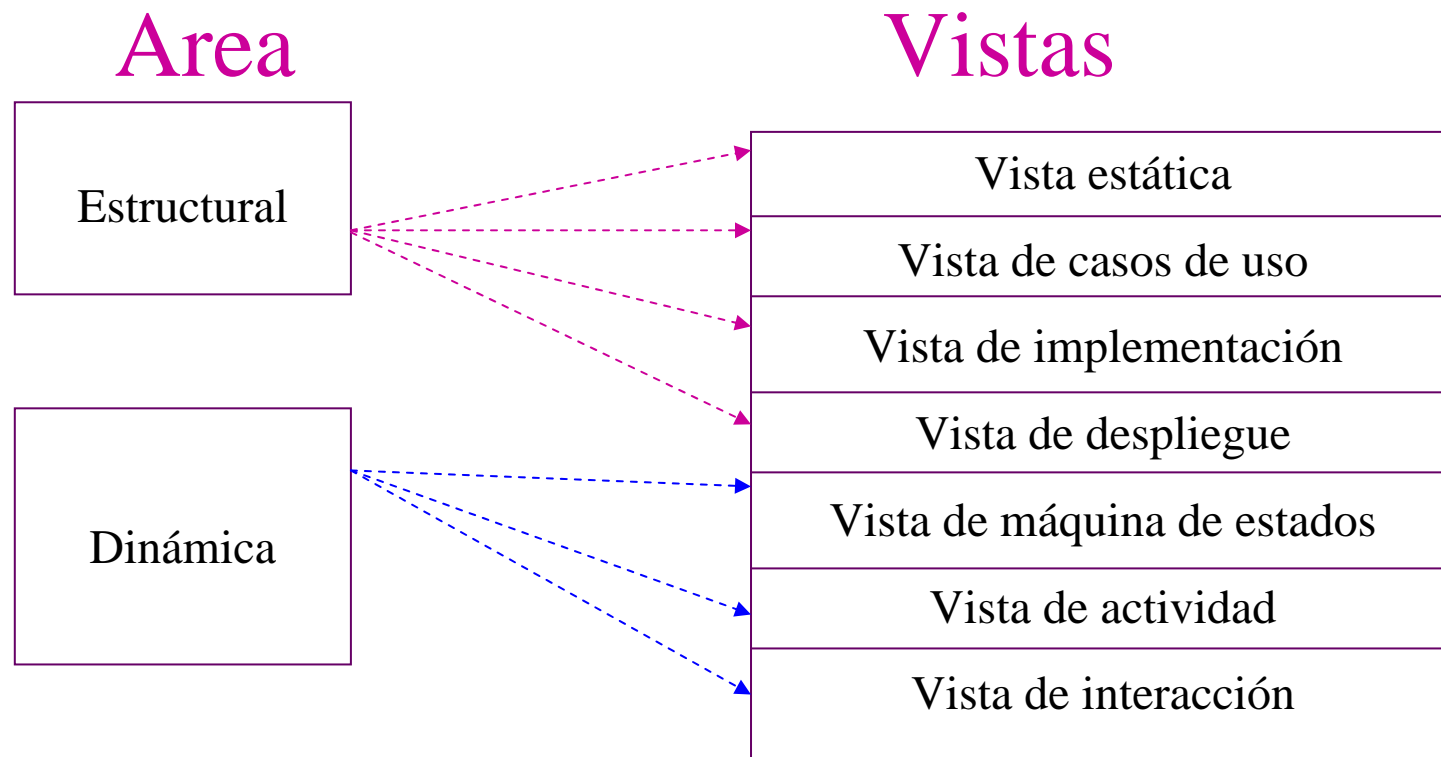
Otros métodos



Lenguaje de modelado

- Para expresar mediante diagramas gráficos diferentes vistas o perspectivas de un sistema en análisis o en desarrollo
- Las vistas en UML
 - Cada vista representa al sistema desde una cierta perspectiva
 - Para cada vista se define un diagrama
 - Son válidas para diferentes enfoques (no necesariamente OO)
 - Una vista es una descripción de un sistema desde una perspectiva particular.

Vistas según UML



Vistas según UML

Área	Vista	Diagramas	Conceptos Principales
estructural	Estática	Clases	Clase, asociación, generalización, ...
	Casos de Uso	Casos de Uso	Caso de uso, actor, asociación, extensión, inclusión, generalización
	Implementación	Componentes	Componente, interfaz, dependencia
	Despliegue	Despliegue	Nodo, componente, dependencia

Vistas según UML

Área	Vista	Diagramas	Conceptos Principales
dinámica	Máquina de estados	Estados	Estado, evento, transición, acción
	Actividad	Actividad	Estado, actividad, transición de terminación, ...
	Interacción	Secuencia	Interacción, objeto, mensaje, activación
		Colaboración	Colaboración, interacción, rol de colaboración, mensaje

Vistas y Diagramas de UML:

Visión estructural

Describe las funcionalidades del sistema a partir de las interacciones del usuario

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

Muestra un conjunto de objetos y sus relaciones en un instante de tiempo

Describe la interacción entre componentes de software

Describe la disposición de instancias de componentes de ejecución en instancias de nodos

Diagrama de Casos de Uso

Diagrama de Clases

Diagrama de Objeto

Casos de Uso

Vista Estática

Diagramas

Diagrama de Componentes

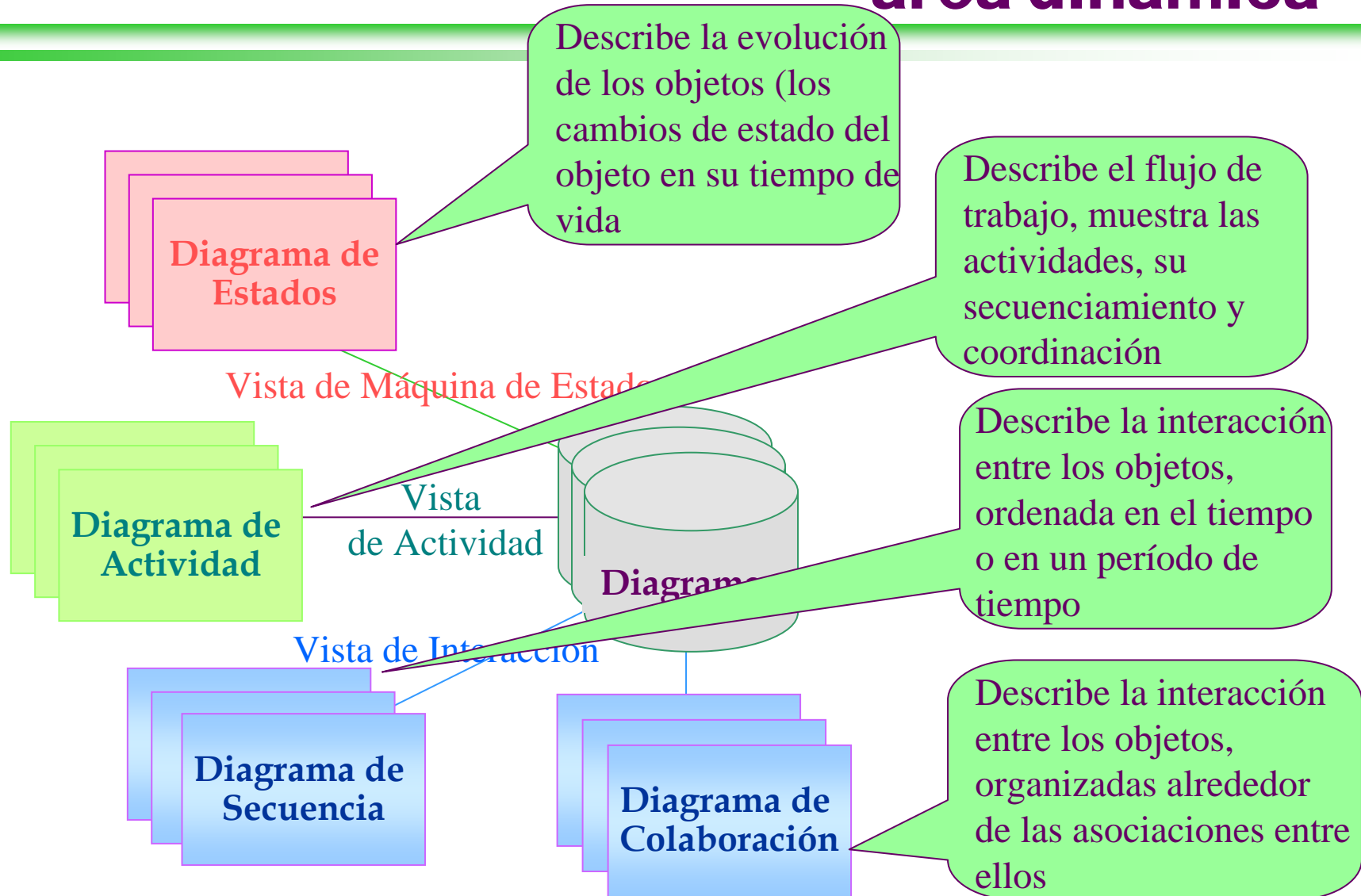
Vista de Implementación y de Despliegue

Diagrama de Despliegue

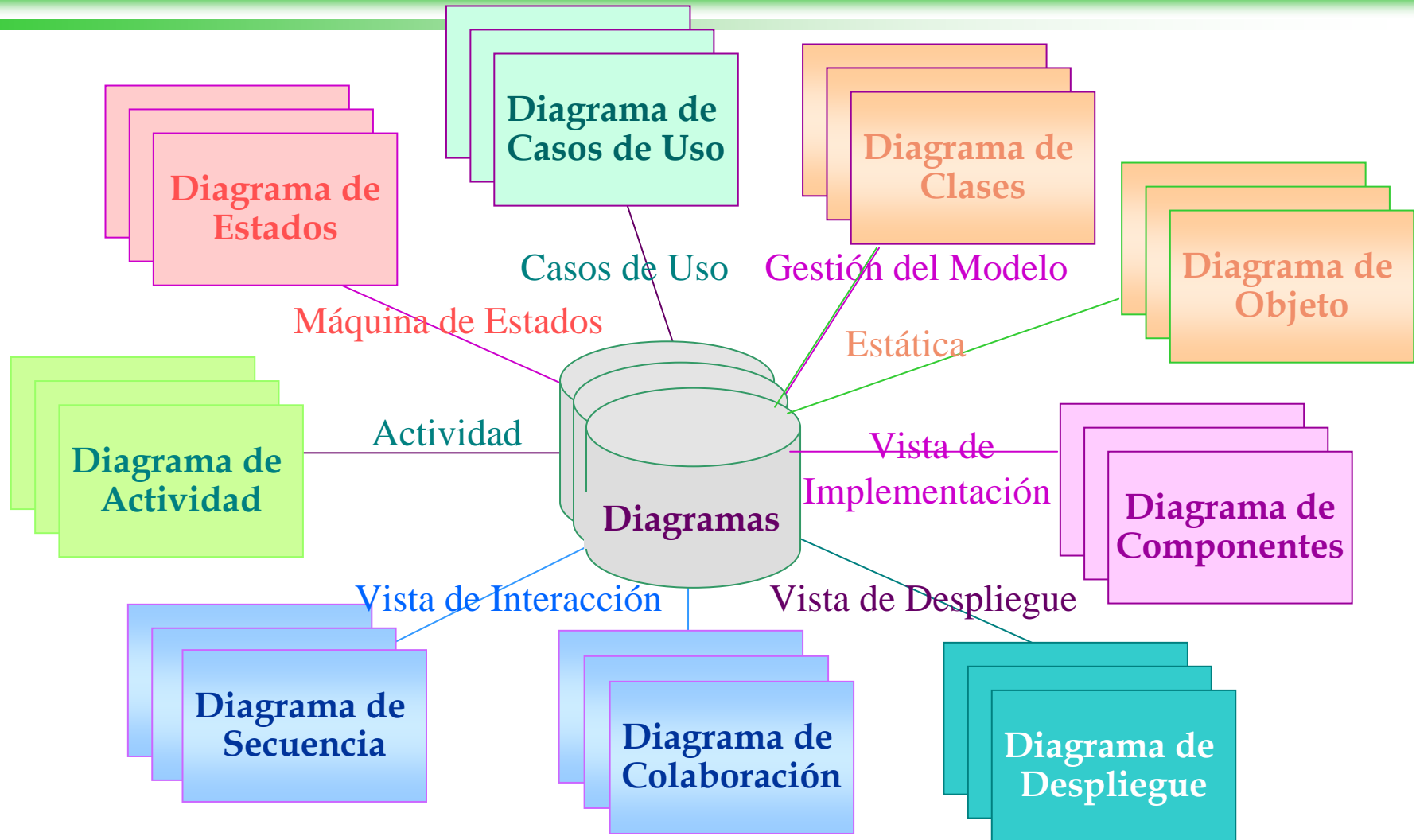


Vistas y Diagramas de UML:

área dinámica



Vistas y Diagramas de UML





Modelo de Caso de Uso

- Describe lo que el sistema debe hacer y bajo que restricciones
- Captura los **requerimientos funcionales** y el ambiente del sistema
- Permite comprender y describir los requerimientos del sistema.

Modelo de Caso de Uso (Componentes)

- **Actor:** Una entidad externa con un comportamiento que **interactúa** con el sistema en desarrollo.
 - Una persona
 - Otro sistema
 - Una organización
- **Usa** el sistema cuando **interactúa** con el **CU**
- **Inicia** la **ejecución** del **CU**.



Modelo de Caso de Uso (Componentes)

- **CU:** Especifica una **secuencia de acciones**, incluyendo sus variantes, que el sistema puede realizar y que **produce un resultado** observable válido para un actor particular.

Modelo de Caso de Uso (Especificación)

- **Diagrama de CU:** Muestra un **conjunto de casos de uso y actores** con una asociación entre cada par actor/caso de uso.

Modelo de Caso de Uso (Relaciones)

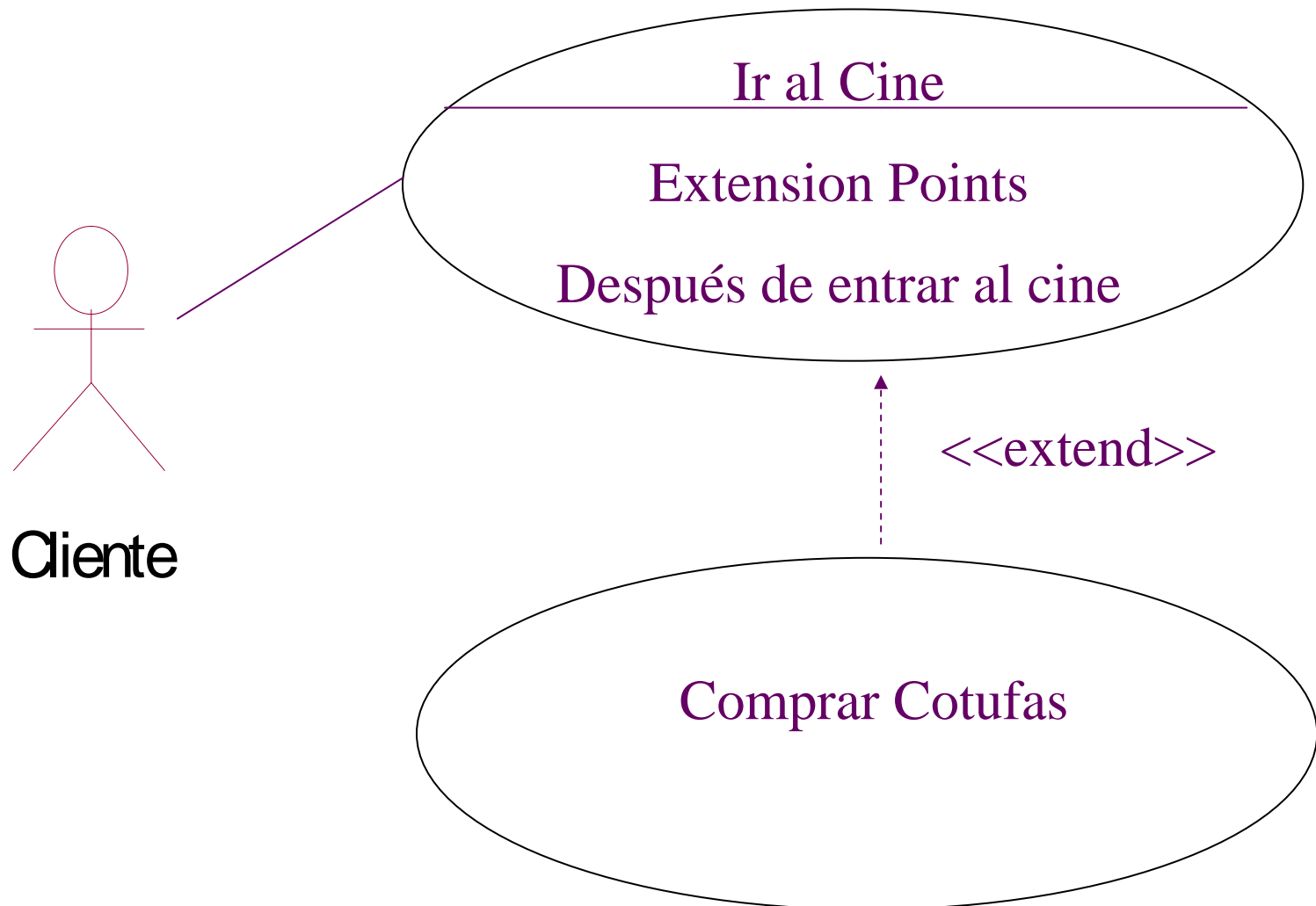
- Relaciones entre actores y casos de uso:
 - **Asociación.**
- Relaciones entre casos de uso:
 - **Extensión (<<extend>>)**
 - **Inclusión (<<include>>)**
 - **Generalización**
- Relaciones entre actores:
 - **Generalización.**



Modelo de Caso de Uso (Relación Extend)

- **Extension Points:** el caso de uso podrá ejecutarse una vez alcanzado el (los) punto de extensión indicado(s).

Ejemplo de Diagrama de Casos de Uso

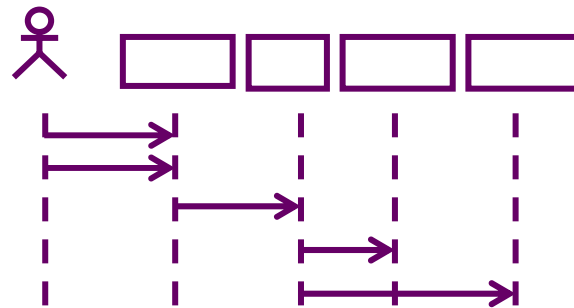




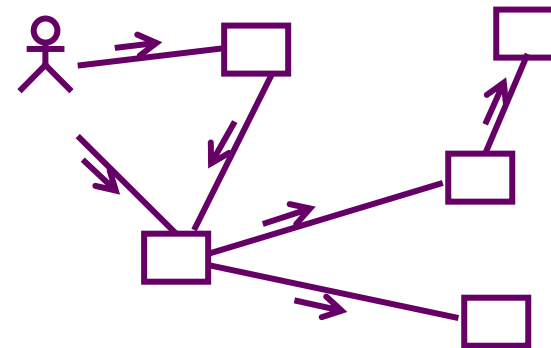
Diagramas de Interacción

- Muestra los objetos que participan en una **interacción** y el orden de los mensajes en el tiempo
- Captura el **comportamiento dinámico**
- Tipos:
 - Diagramas de Secuencia
 - Diagramas de Colaboración

Ejemplo de: Diagrama de Interacción



Diagramas de
Secuencia



Diagramas de
Colaboración



Diagramas de Secuencia

- Un Diagrama de Secuencia describe la interacción entre objetos haciendo énfasis en el orden de los mensajes en el tiempo
- Maneja la comunicación entre el sistema y el mundo exterior
- Son un medio para clarificar escenarios.

Notación Básica

- ¿Qué representa cada figura?

C

Una clase

o

Un objeto

:C2

Un objeto

o:C

Un objeto

Escenario principal de éxito

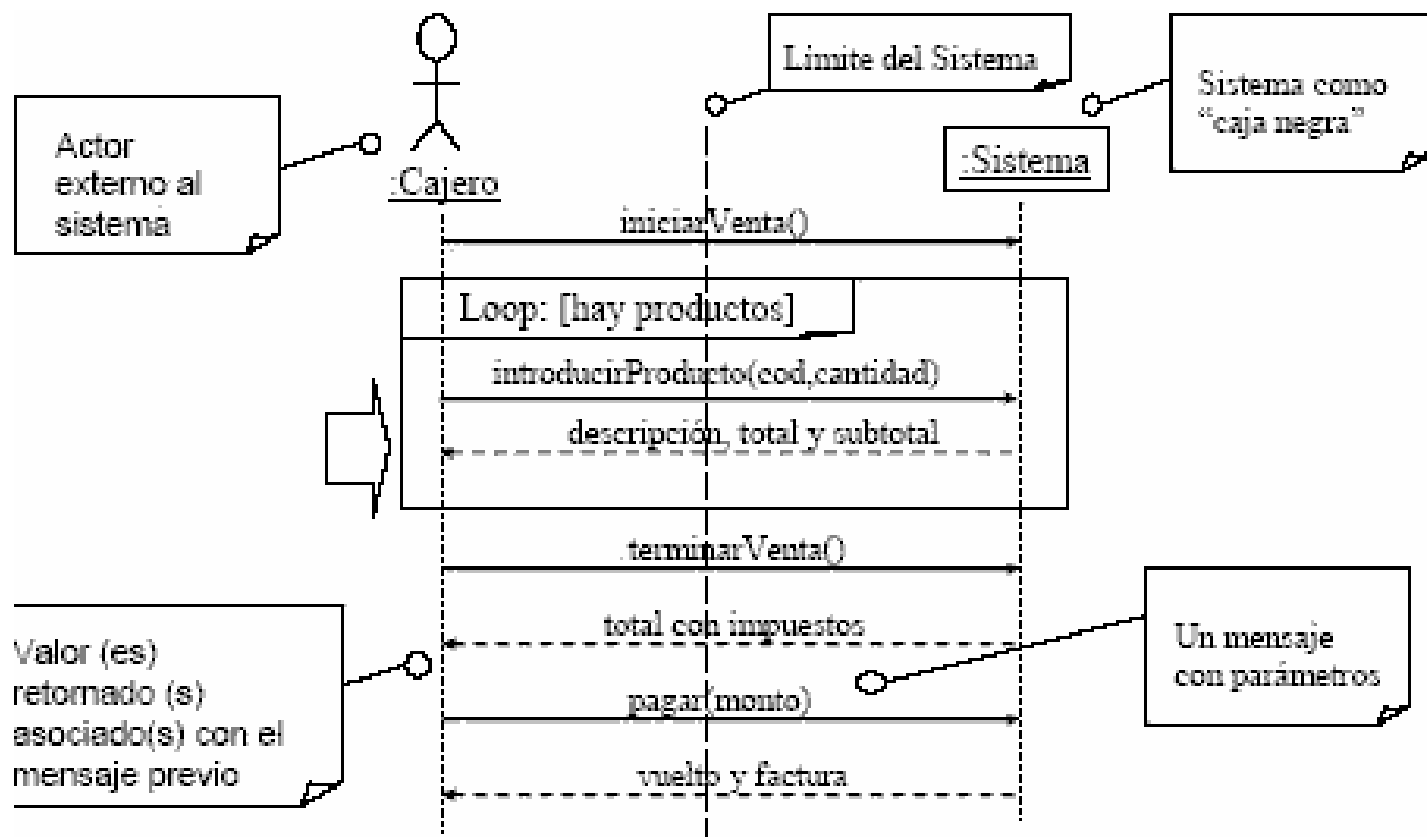
Diagrama de Secuencia de Procesar Venta

Escenario Principal de Éxito (pago en efectivo solamente):

1. El Cliente llega a la caja con los artículos y servicios a pagar
 2. El Cajero inicia una nueva venta
 3. El Cajero introduce el código del producto
 4. El Sistema registra el producto y presenta la descripción del producto, precio y subtotal
- El cajero repite los pasos 3-4 hasta procesar todos los artículos y servicios*
5. El Sistema muestra el total con el impuesto calculado
 6. El Cajero indica al Cliente el total y pregunta la forma de pago
 7. El Cliente paga y el Sistema maneja el pago
 8. El Sistema registra la venta
 9. El Sistema emite la factura
 10. El Cliente se marcha con lo adquirido y la factura

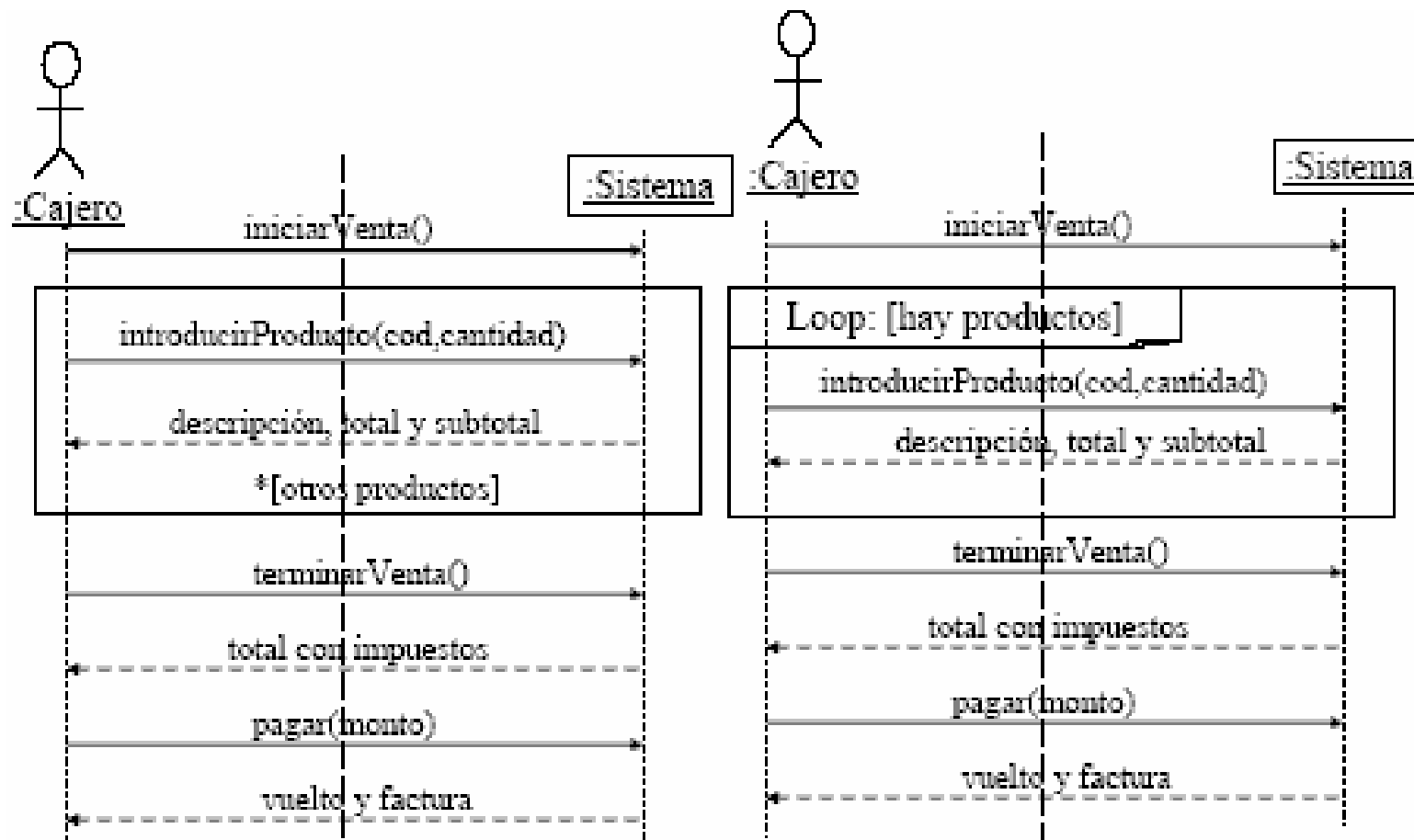
Escenario principal de éxito

Diagrama de Secuencia de Procesar Venta

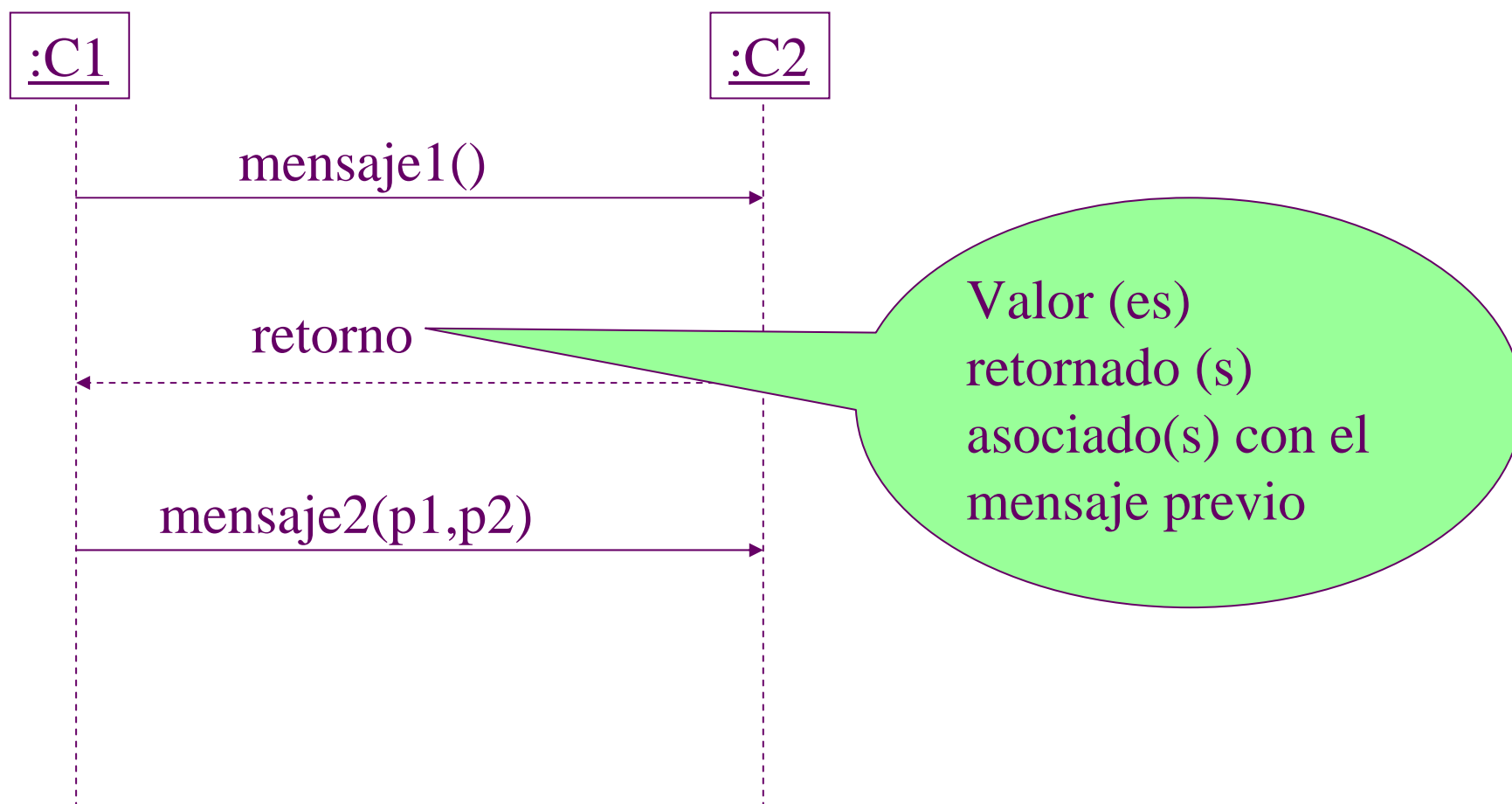


Escenario principal de éxito

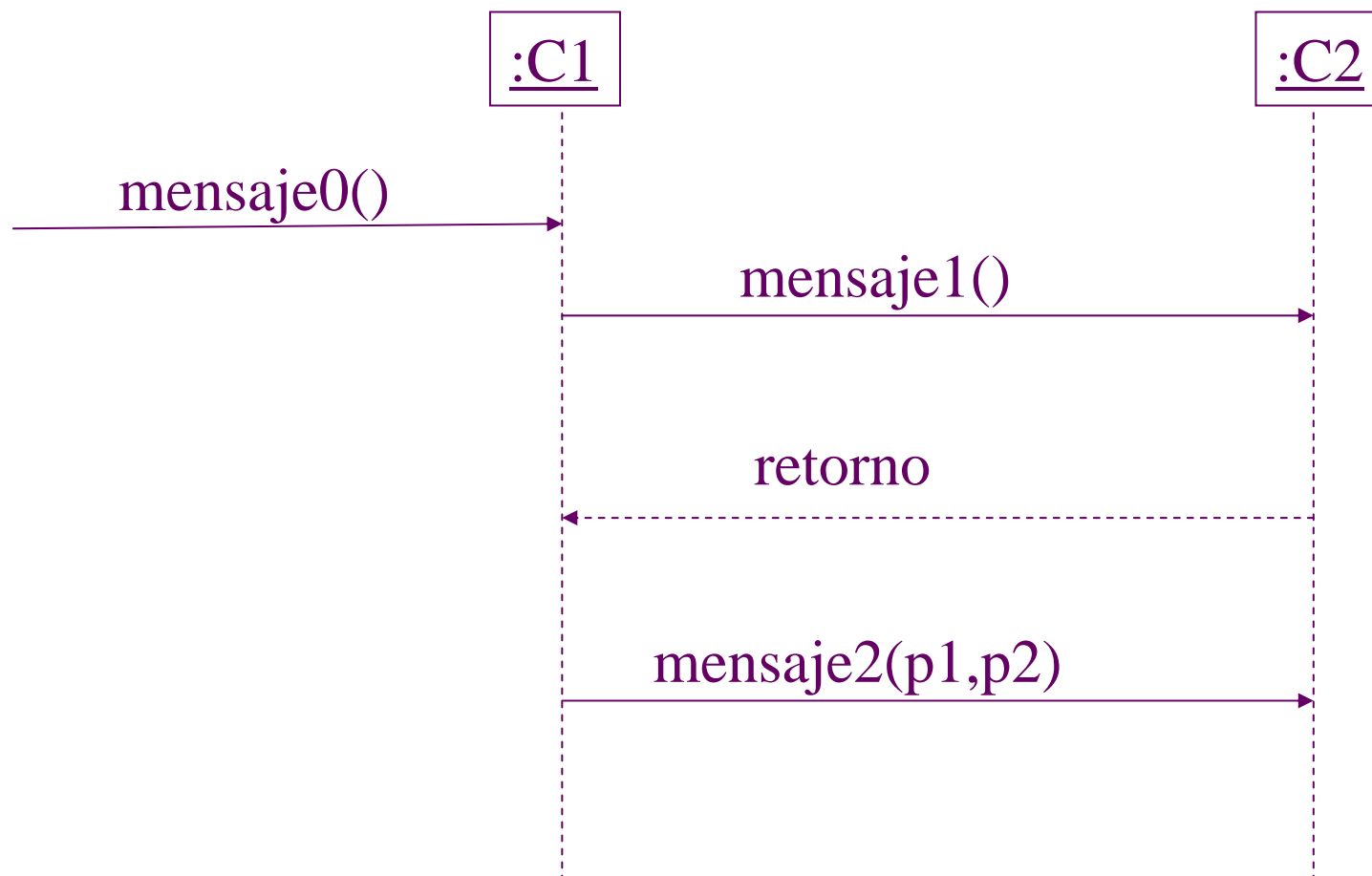
Diagrama de Secuencia de Procesar Venta



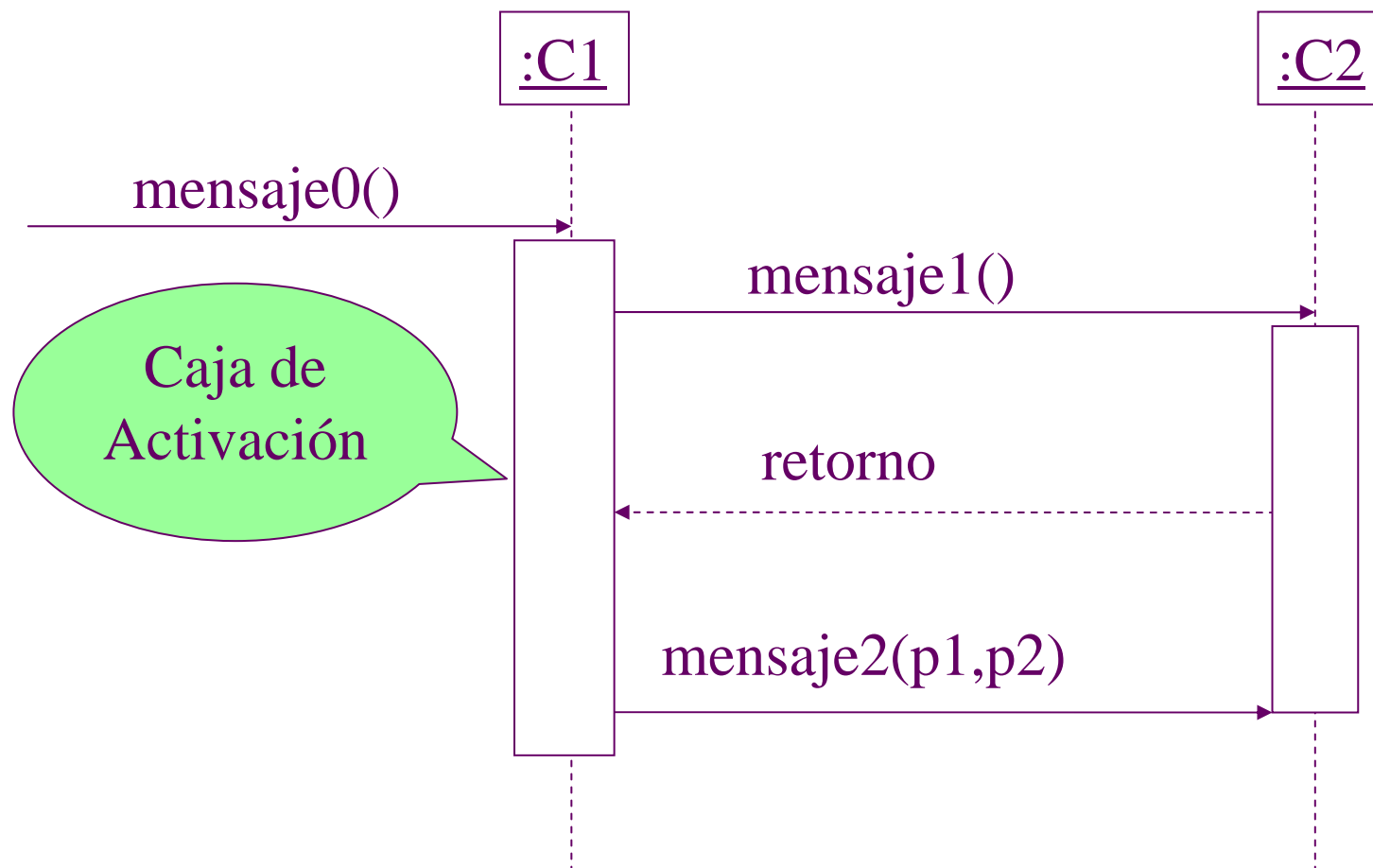
Notación Básica



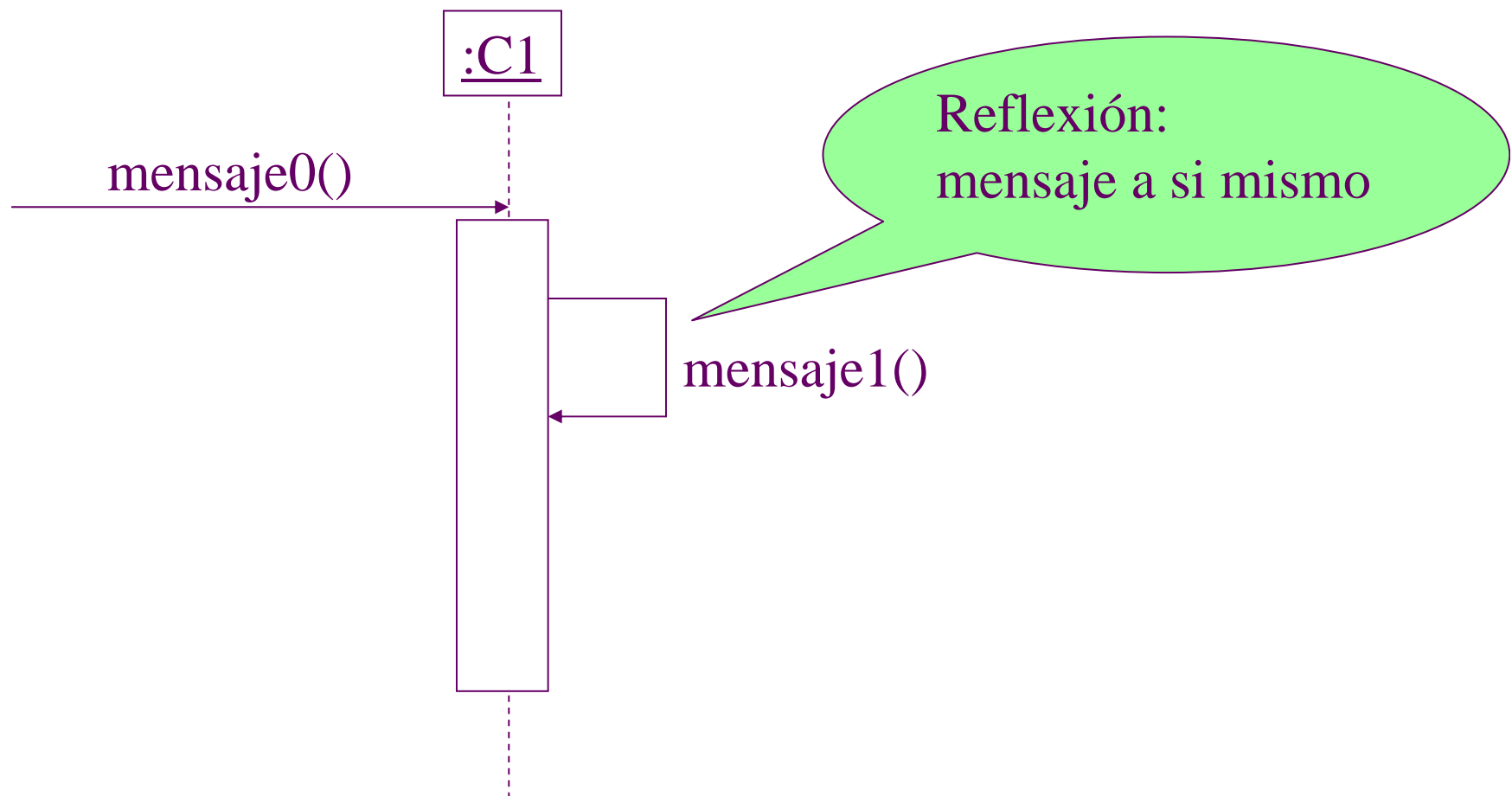
Notación Básica



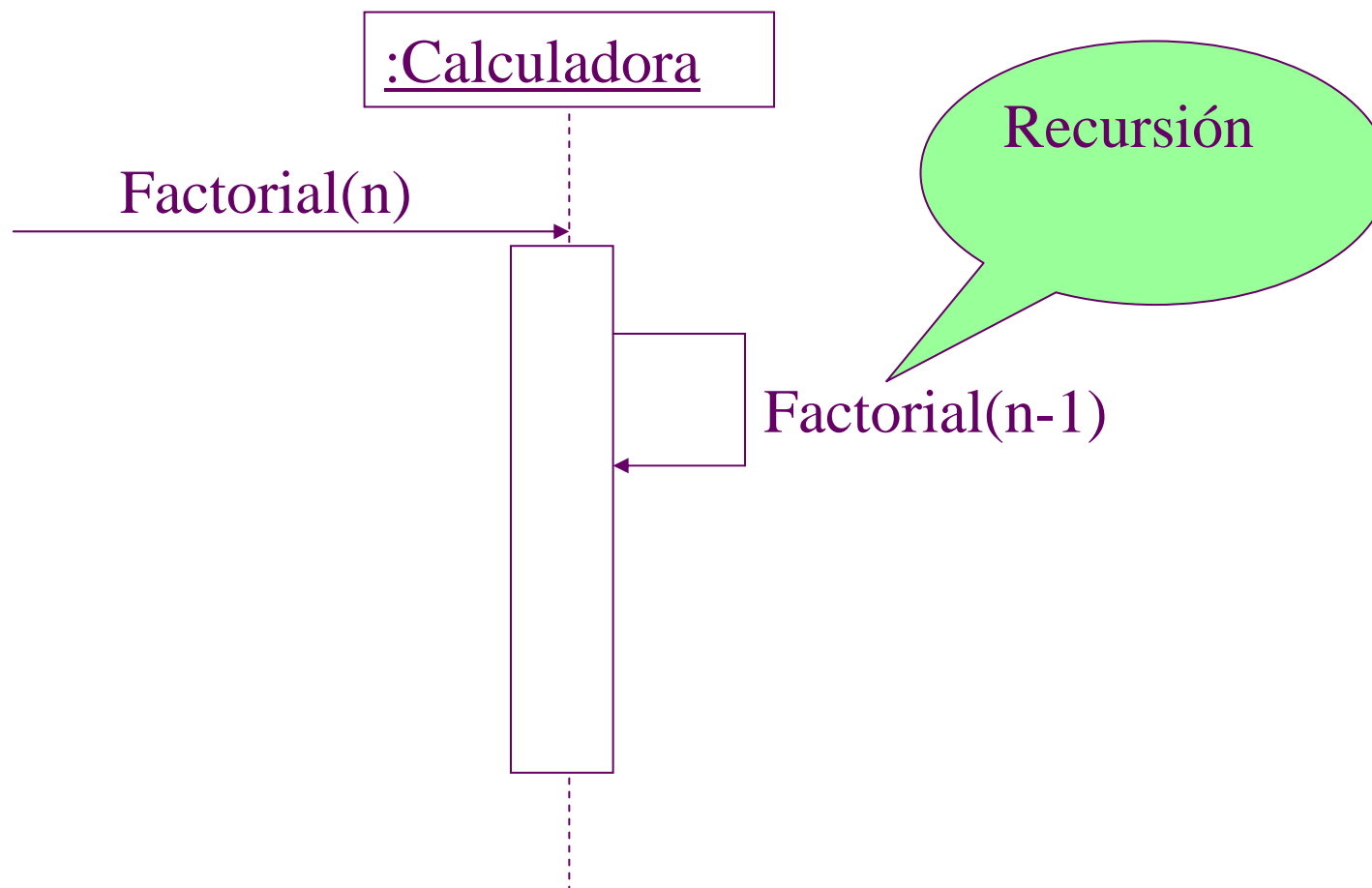
Notación Básica



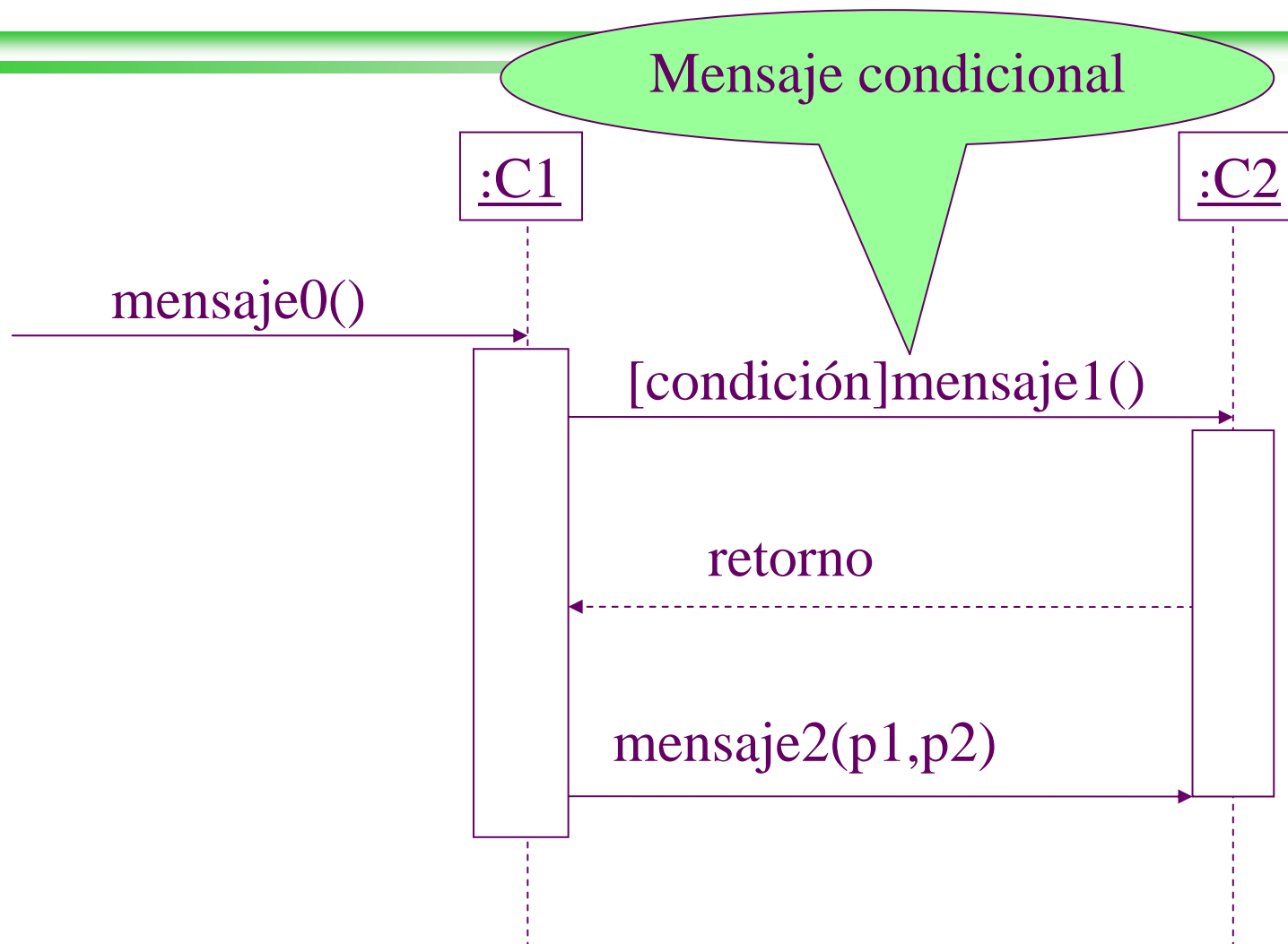
Notación Básica



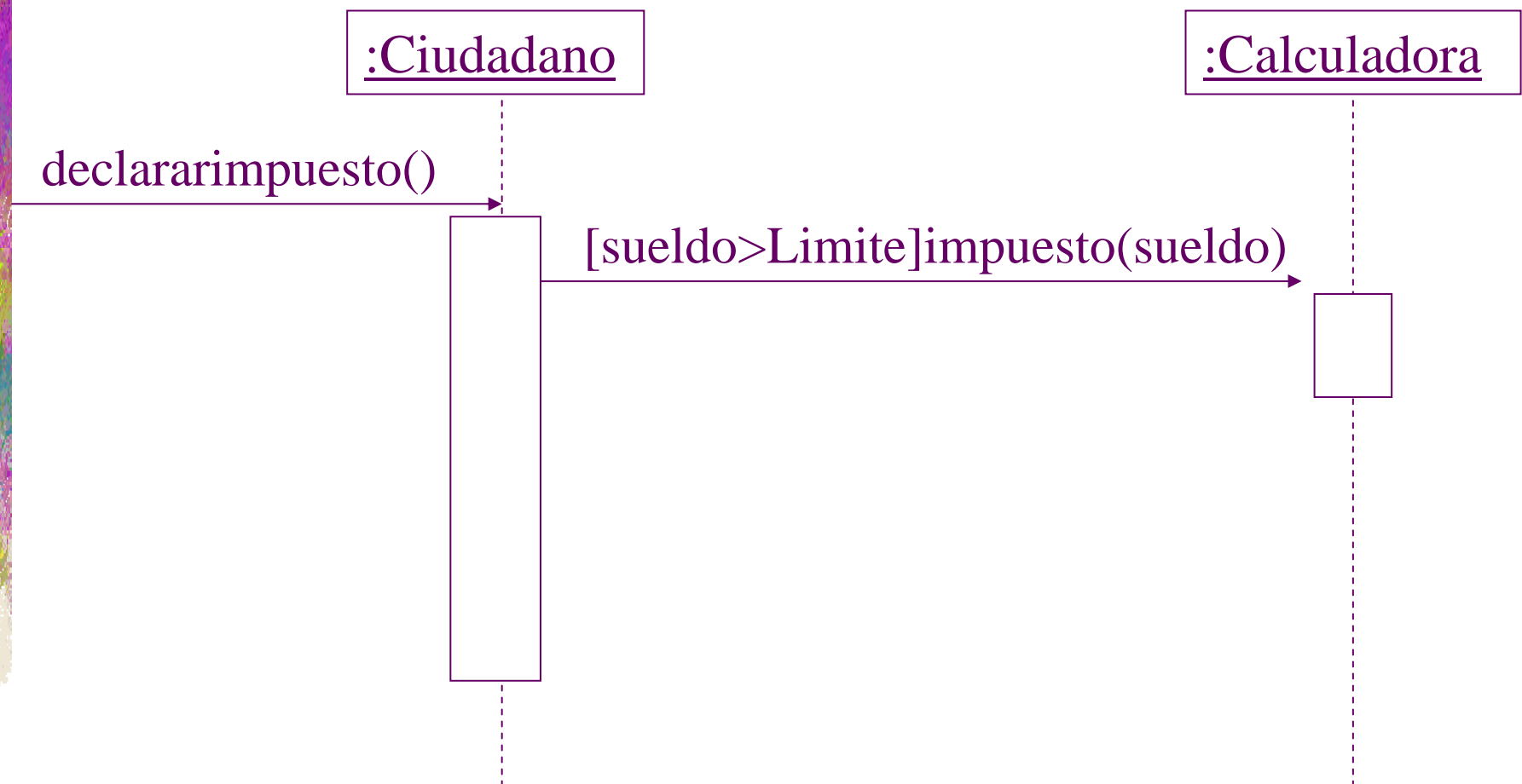
Notación Básica



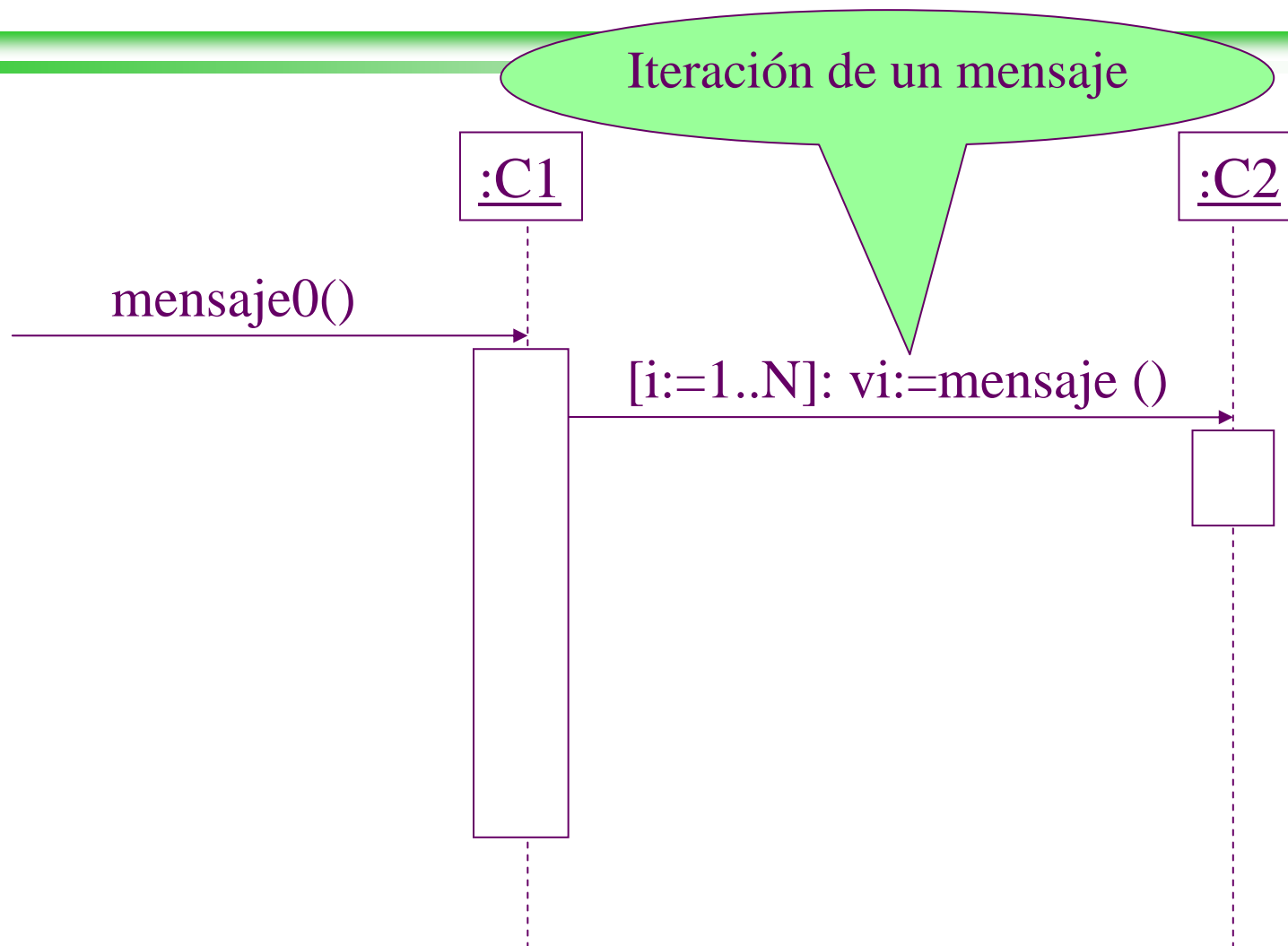
Notación Básica



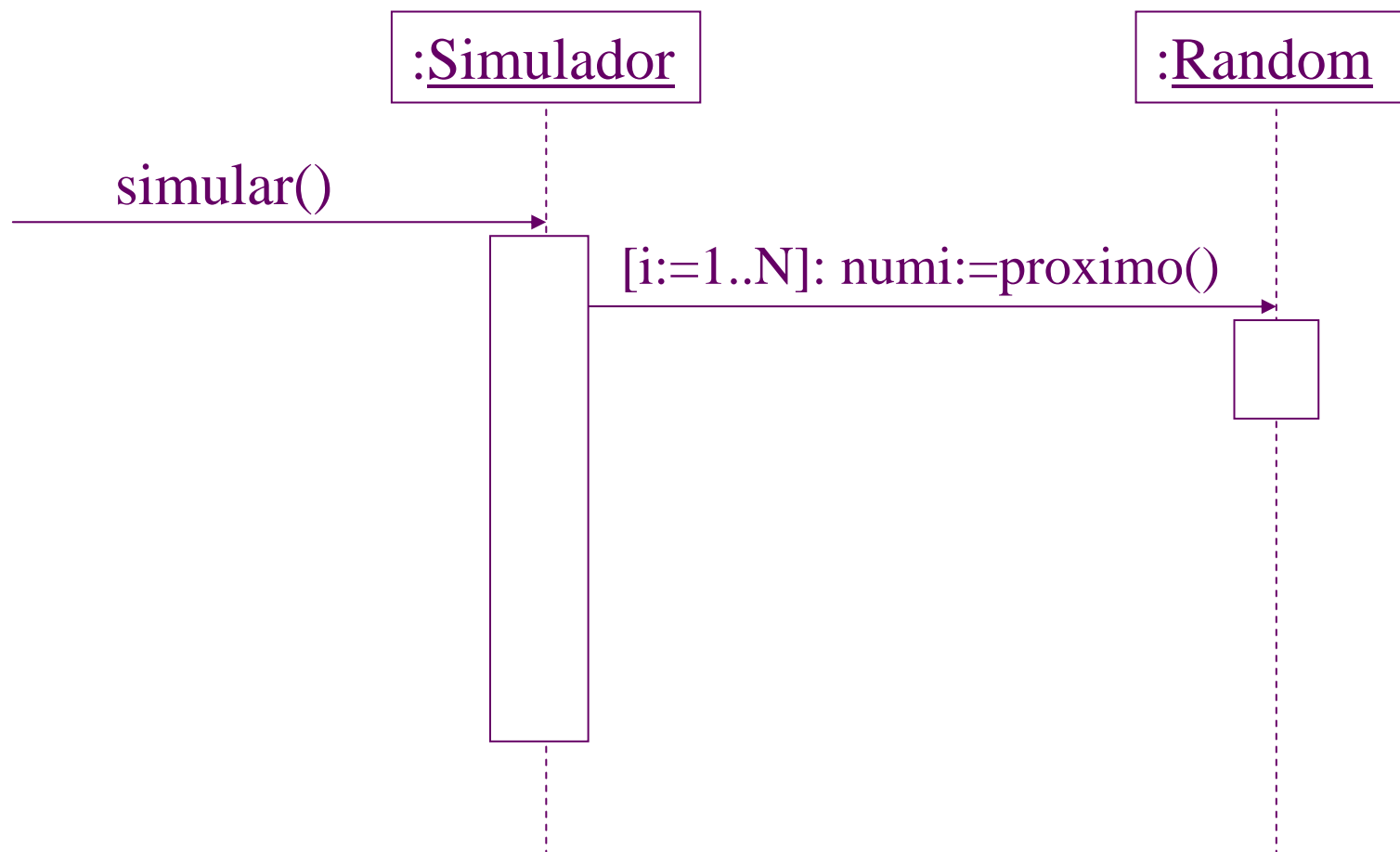
Notación Básica



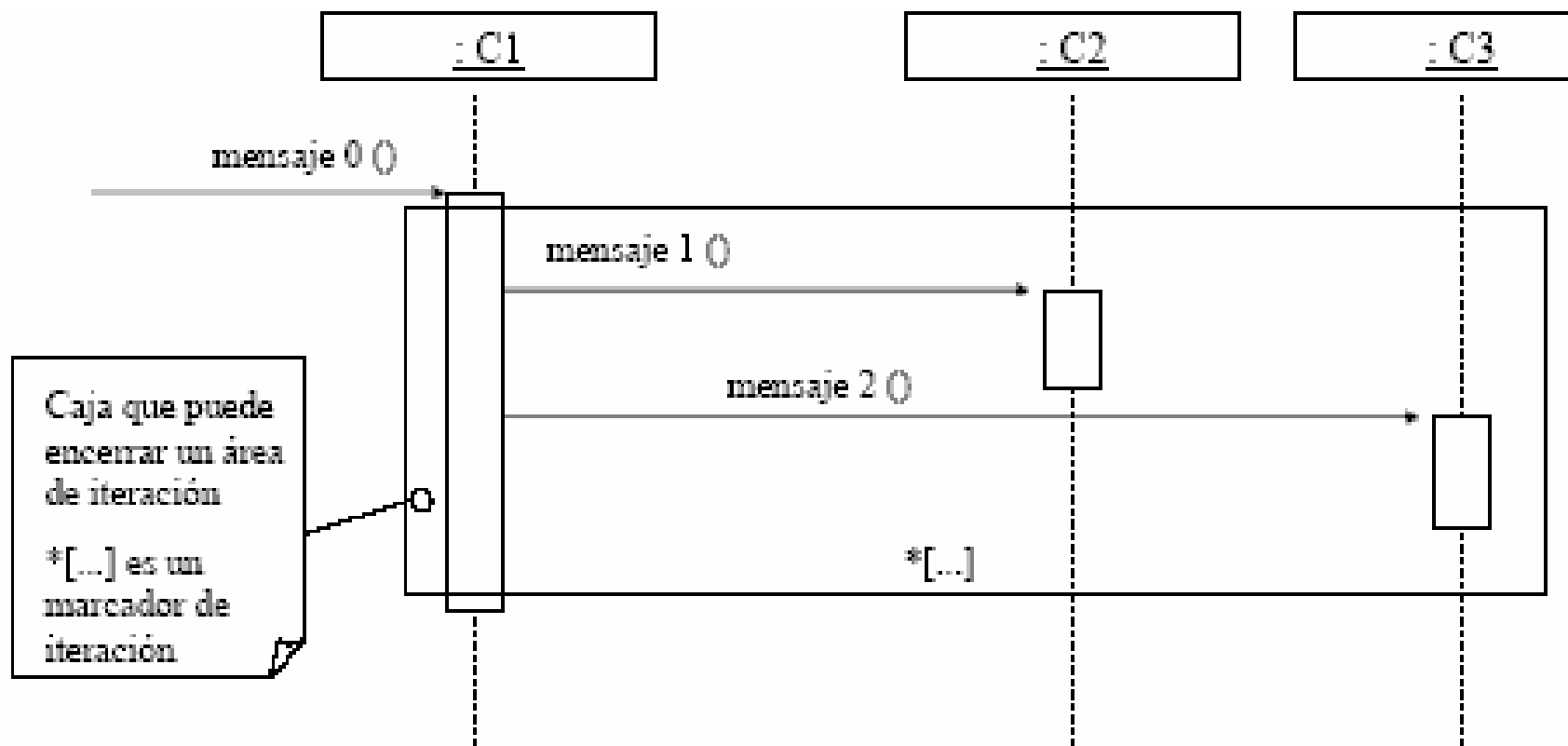
Notación Básica



Notación Básica



Notación Básica

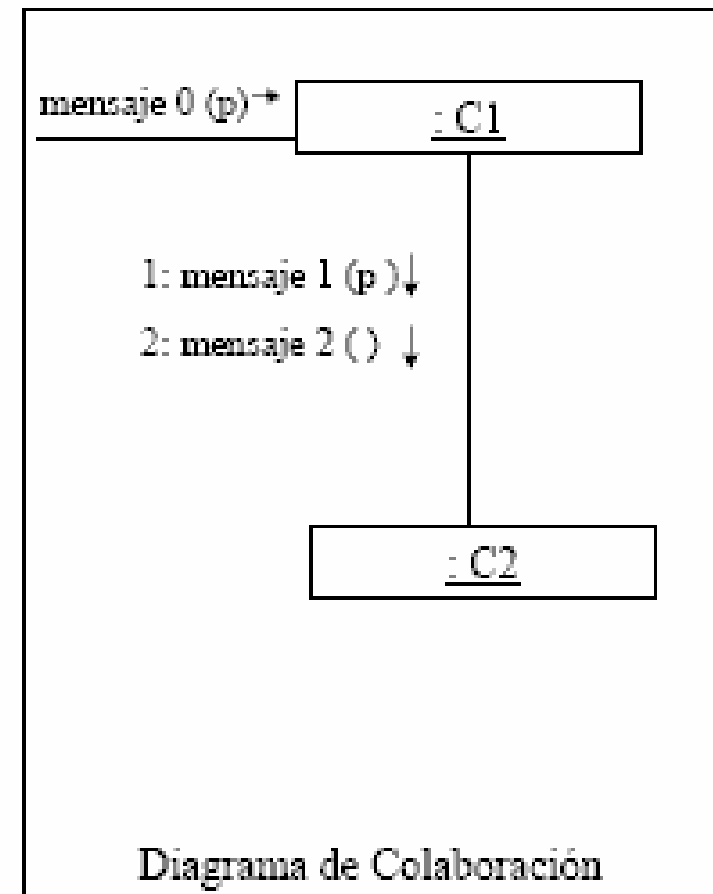
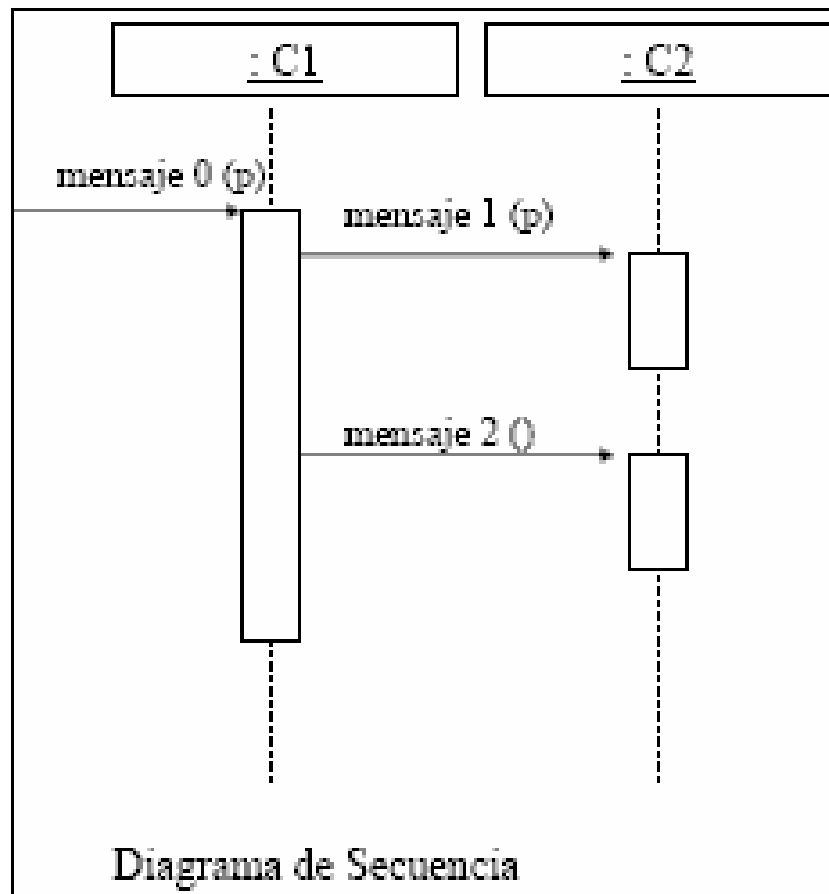




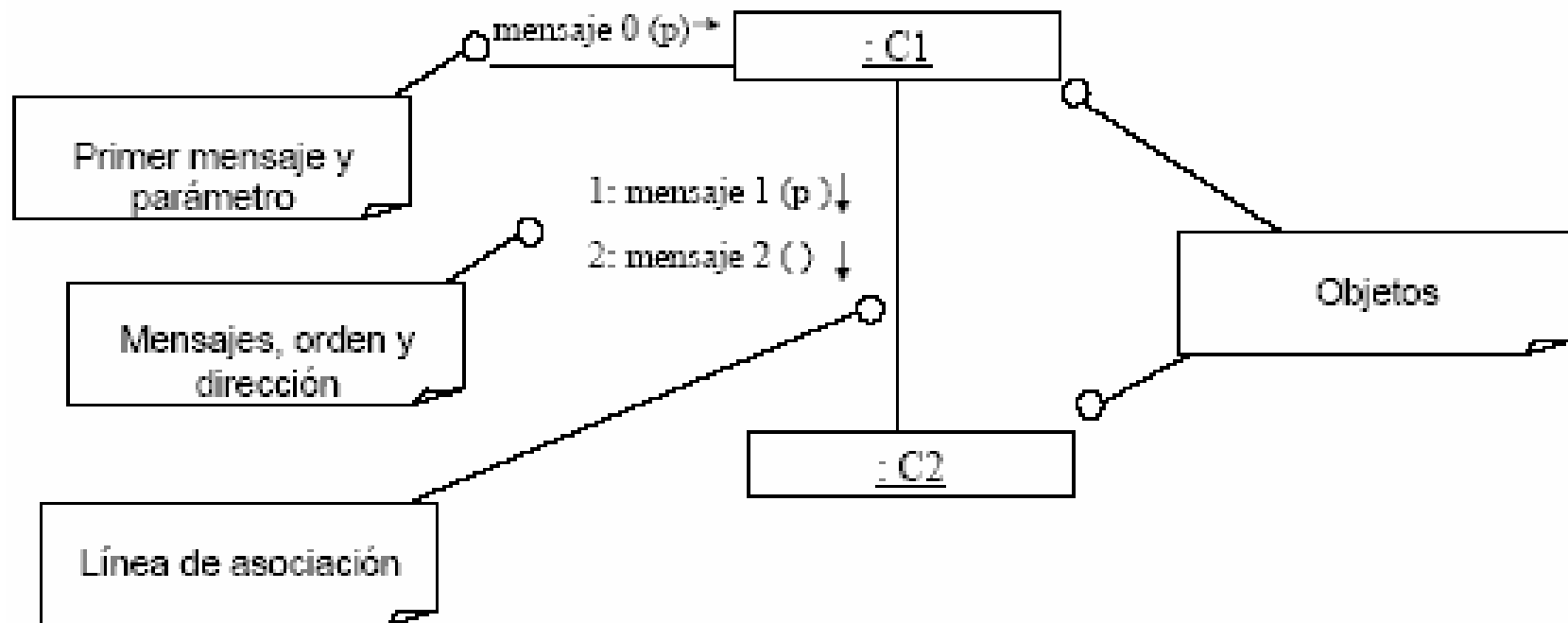
Diagramas de Colaboración

- Un Diagrama de Colaboración describe las interacciones entre objetos haciendo énfasis en la estructura de la colaboración

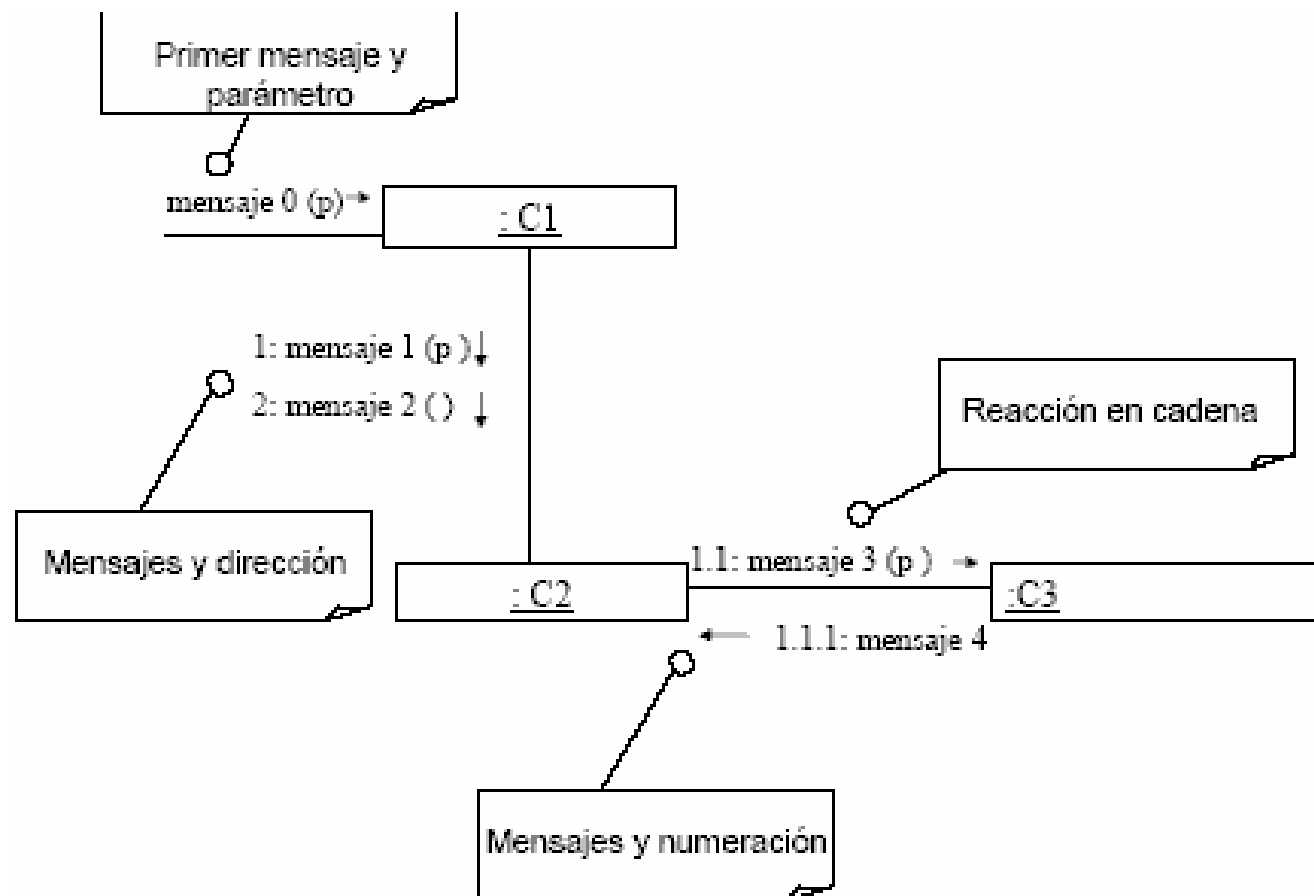
Diagramas de Colaboración



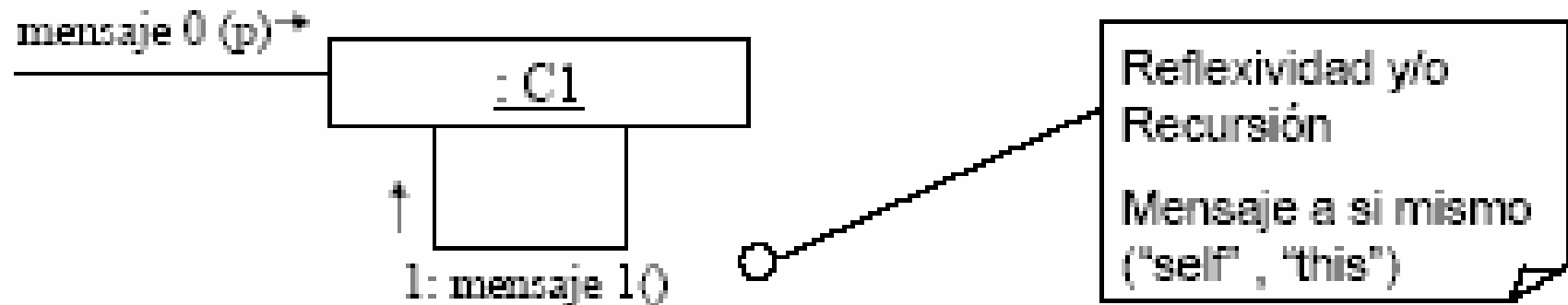
Diagramas de Colaboración



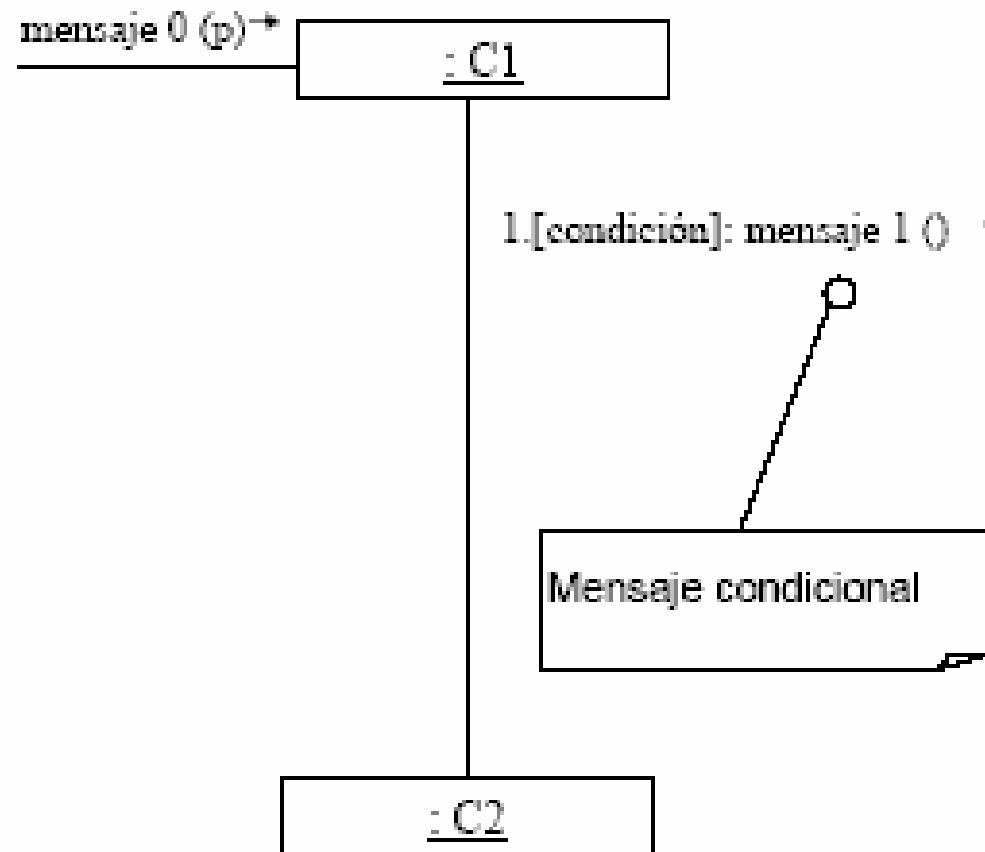
Diagramas de Colaboración



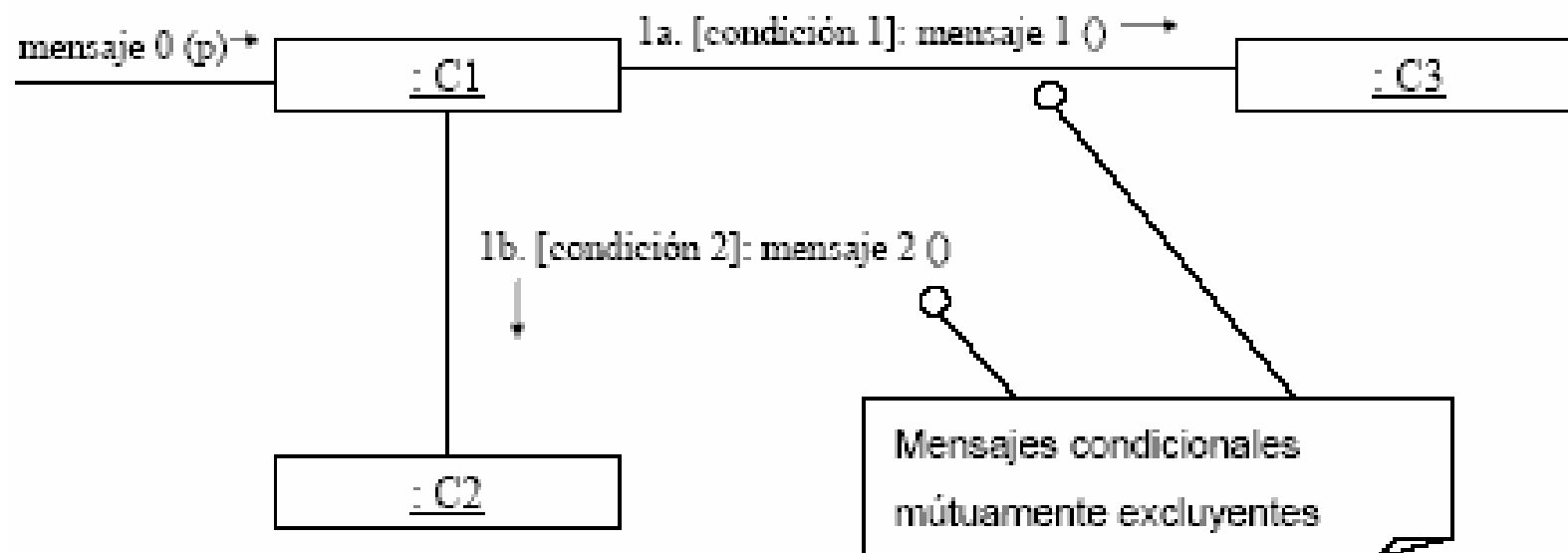
Diagramas de Colaboración



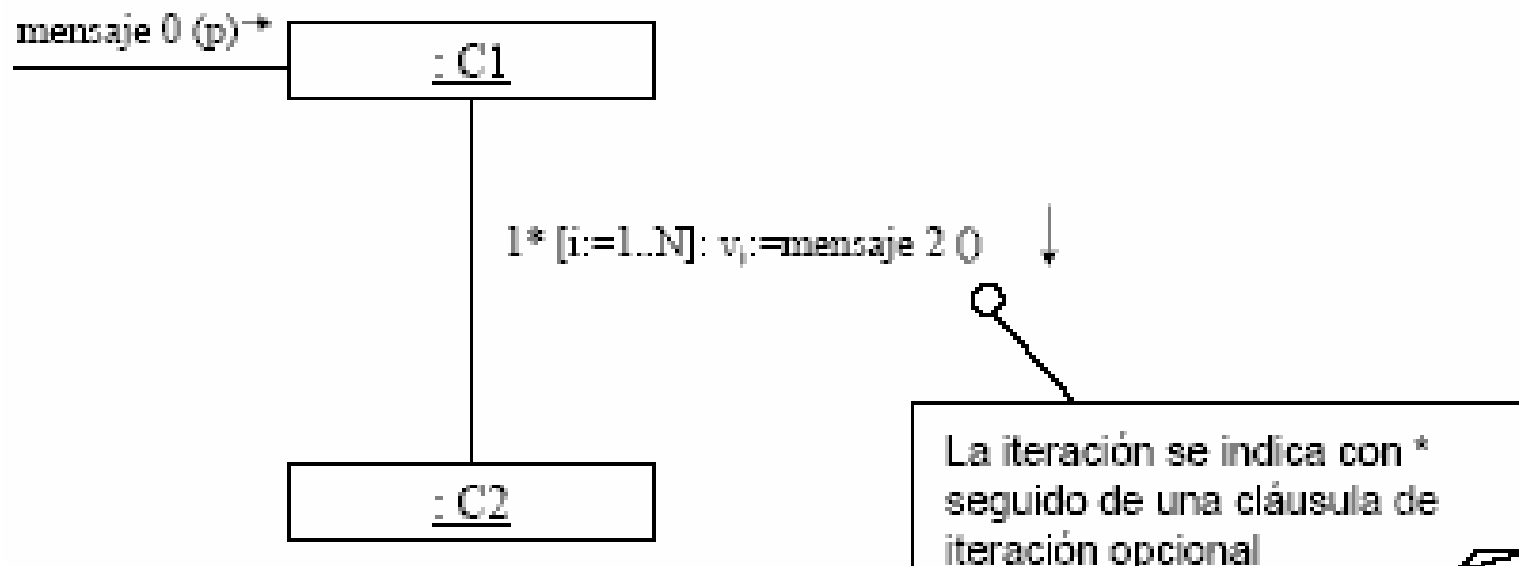
Diagramas de Colaboración



Diagramas de Colaboración



Diagramas de Colaboración





Diagramas de Secuencia UML 2.0

“UML’s Sequence Diagram”

Donald Bell

IBM

12 January 2004

Diagramas de Secuencia UML 2.0

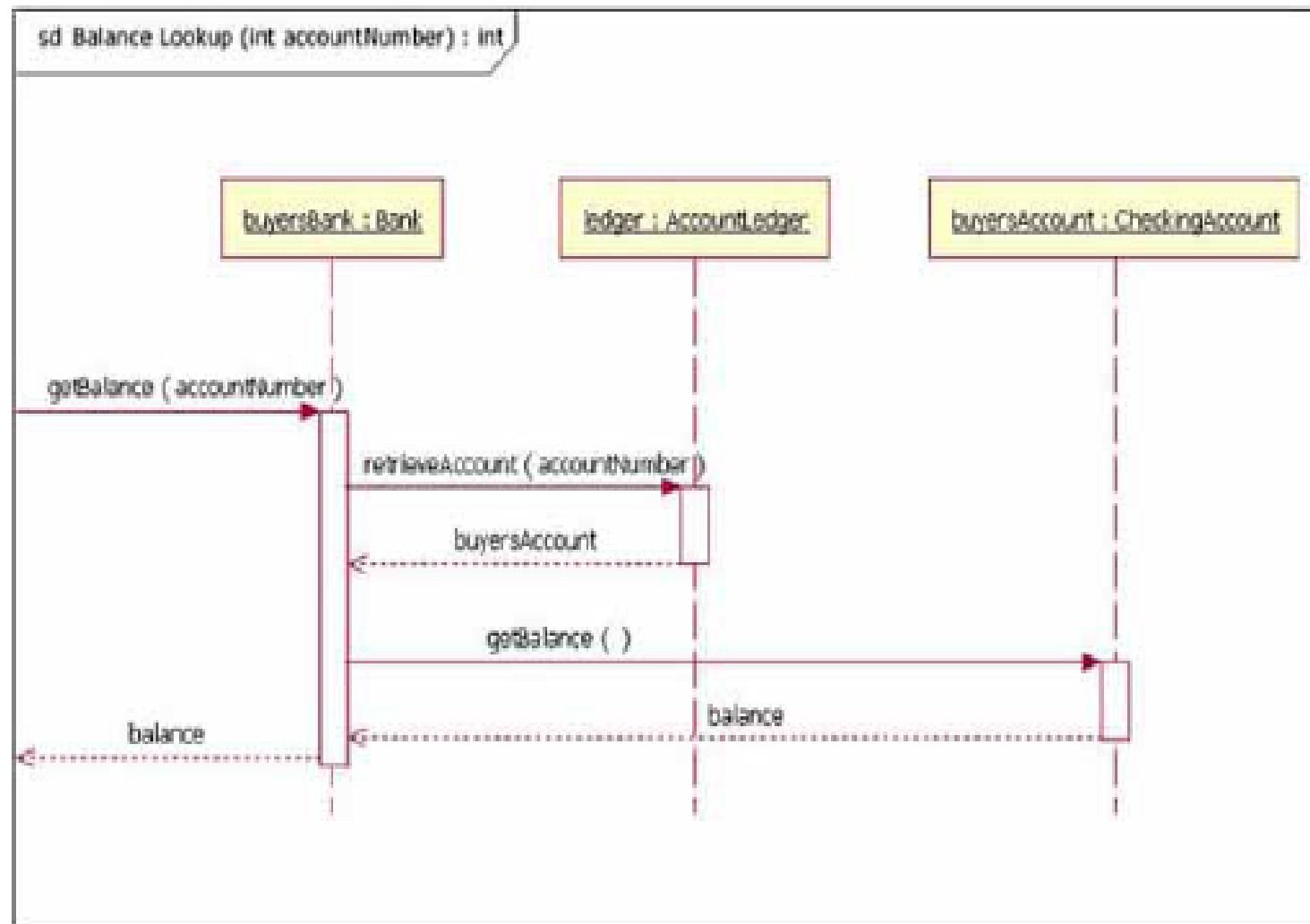
<Diagram Type> <Diagram Name>

- sd=Sequence Diagram,
- activity=Activity Diagram,
- use case=Use Case Diagram

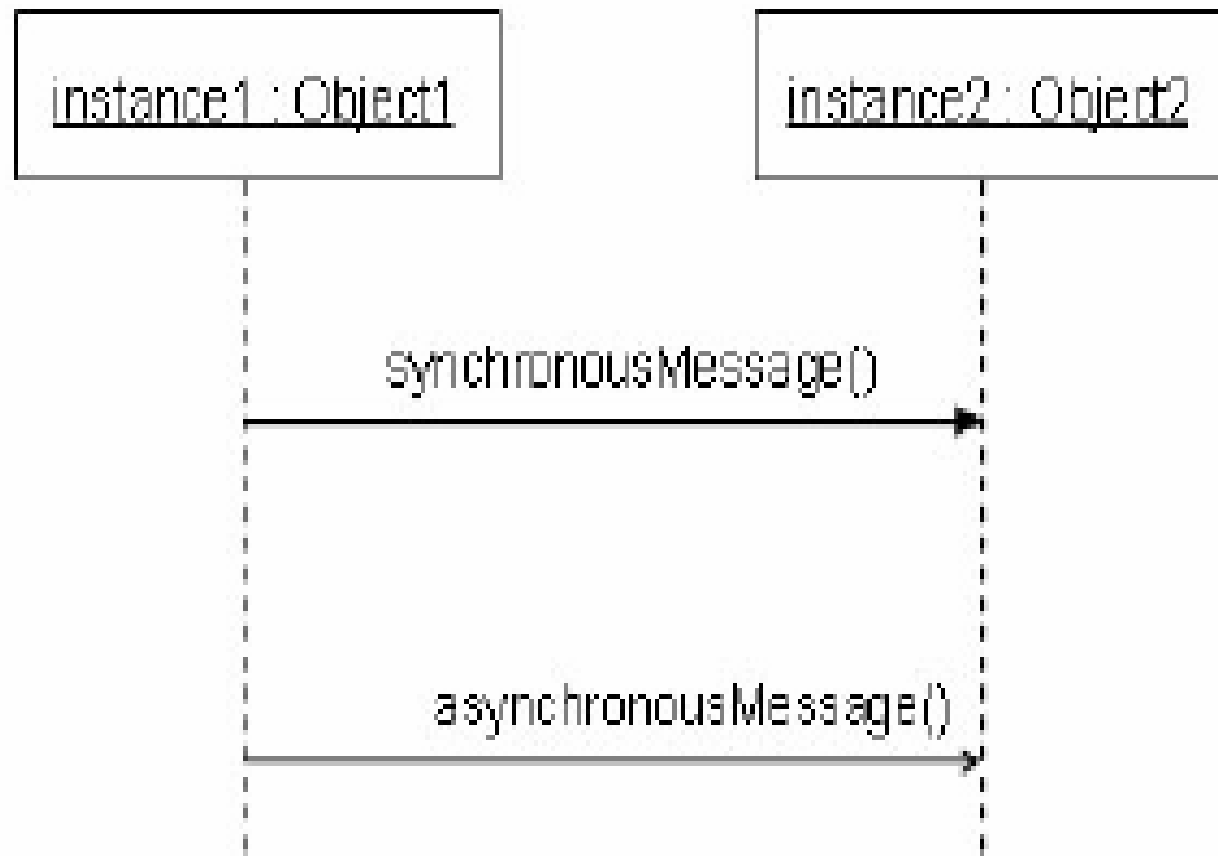
< Diagram's Label >

< Diagram's Content Area >

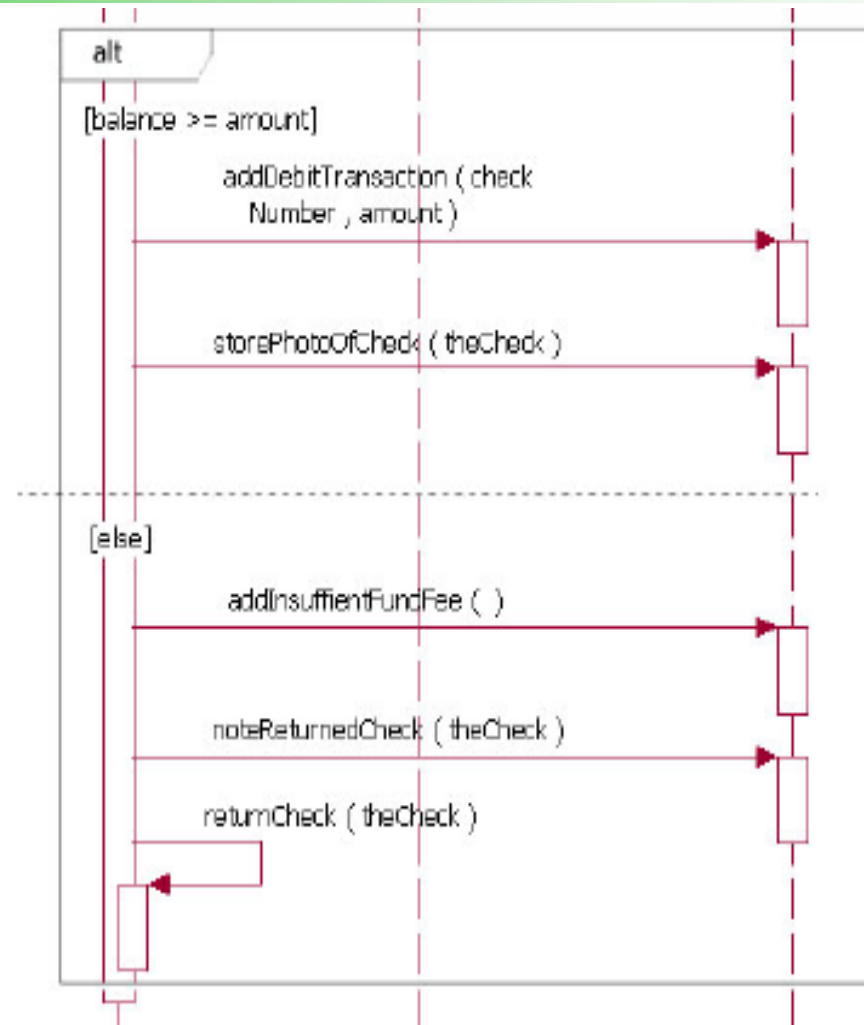
Diagramas de Secuencia UML 2.0



Mensajes asíncronos



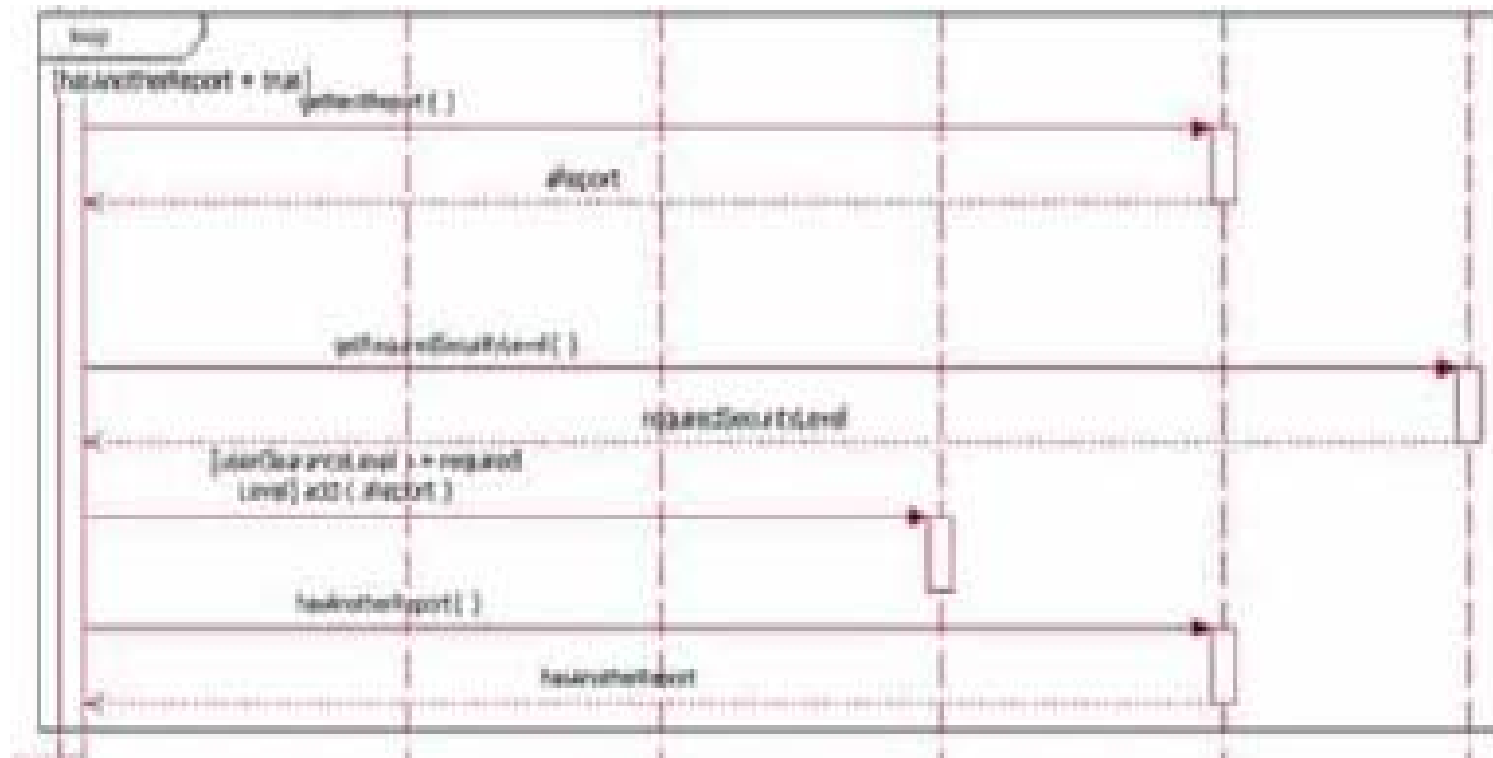
Alternativas: If ...else...



Opciones: If



Loops





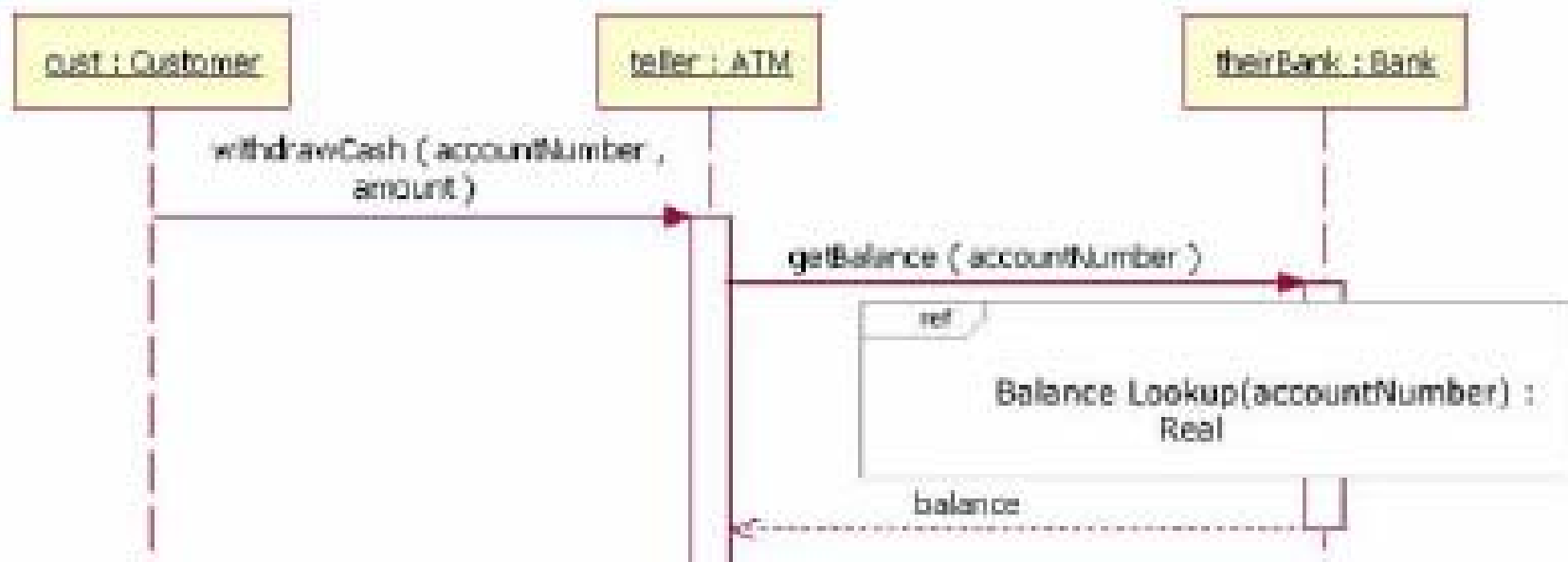
Características Avanzadas

- Referenciando otros diagramas de secuencia
- Break
- Tareas paralelas

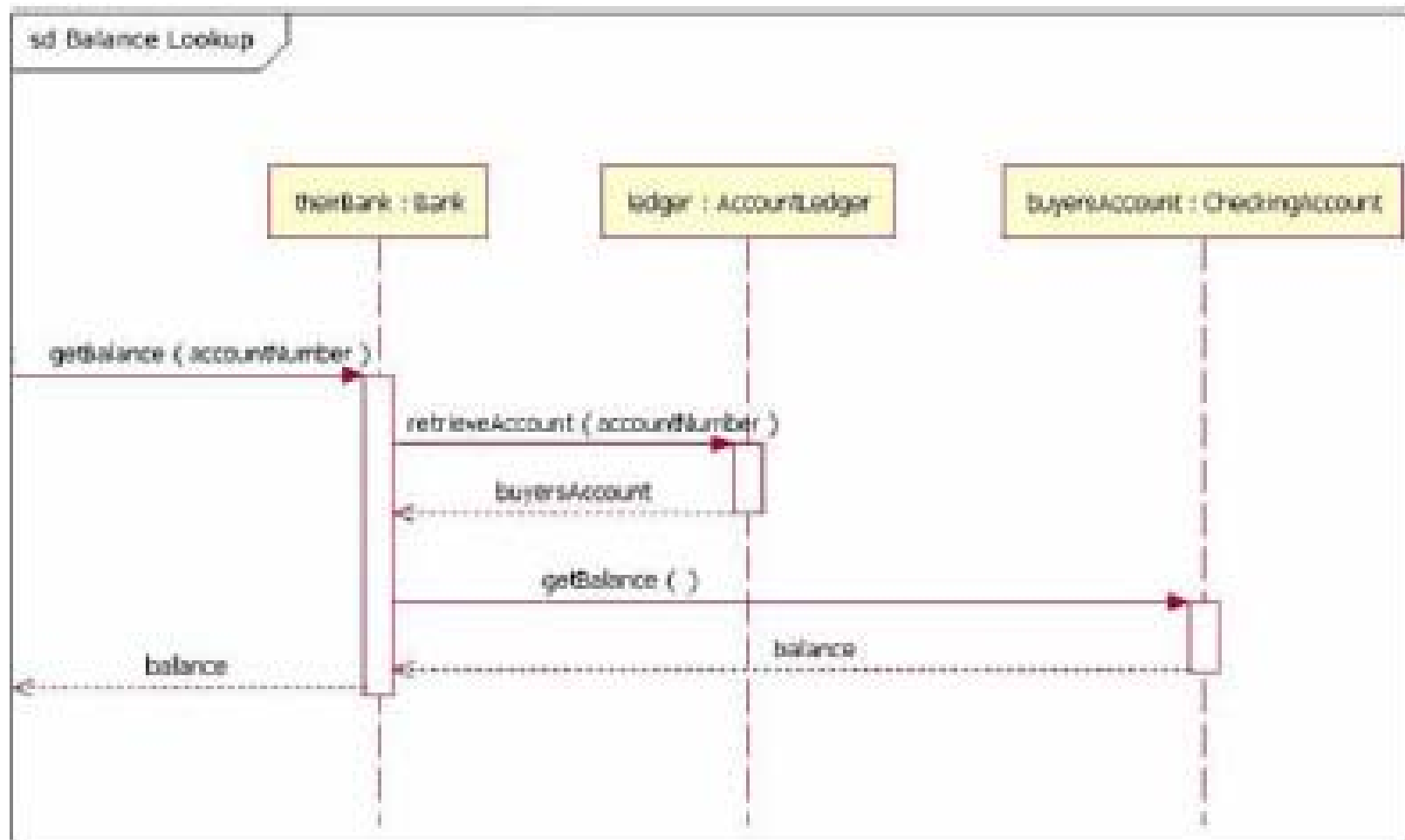
Referencias a diagramas de secuencias

- Notación

***sequence diagram name[(arguments)]
[: return value]***

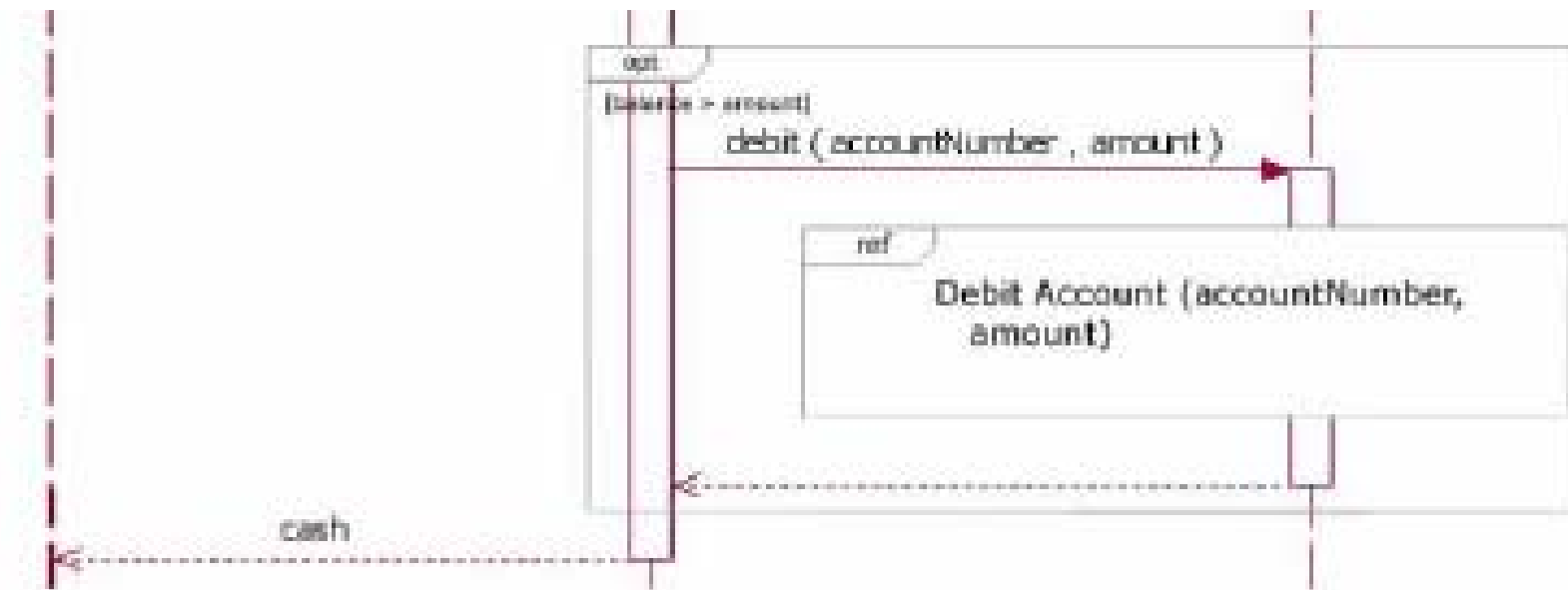


Referencias a diagramas de secuencias

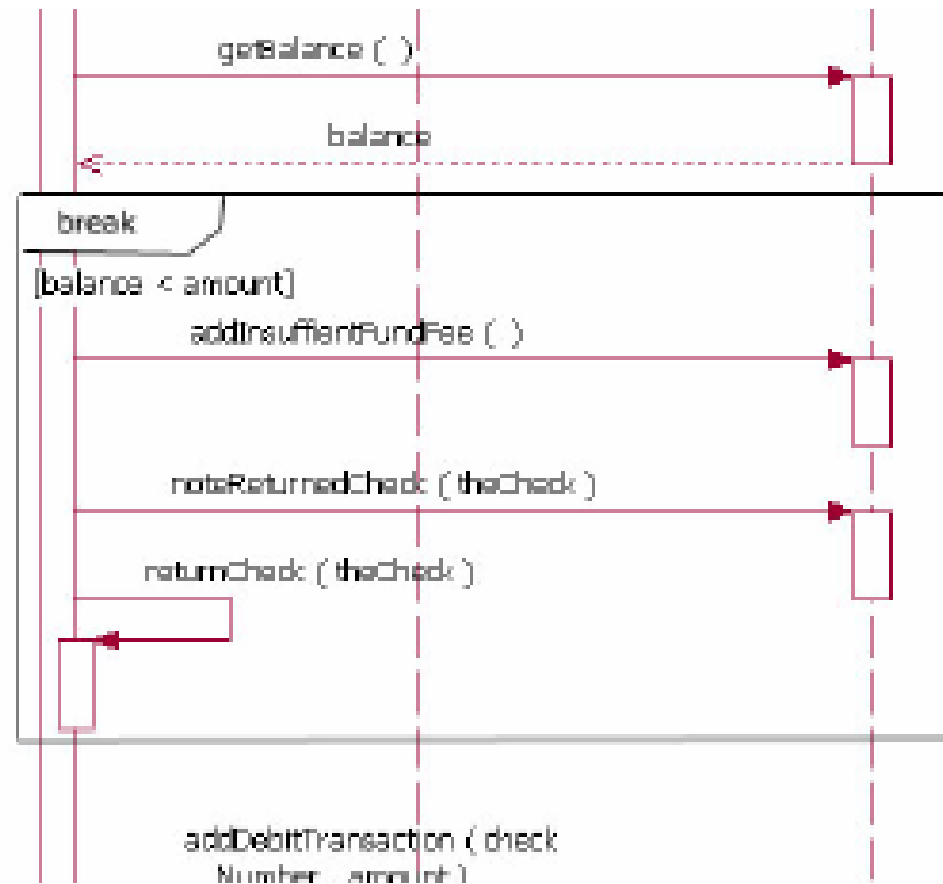


Referencias a diagramas de secuencias

- Notación
sequence diagram name[(arguments)]
[: return value]



Break



Tareas paralelas

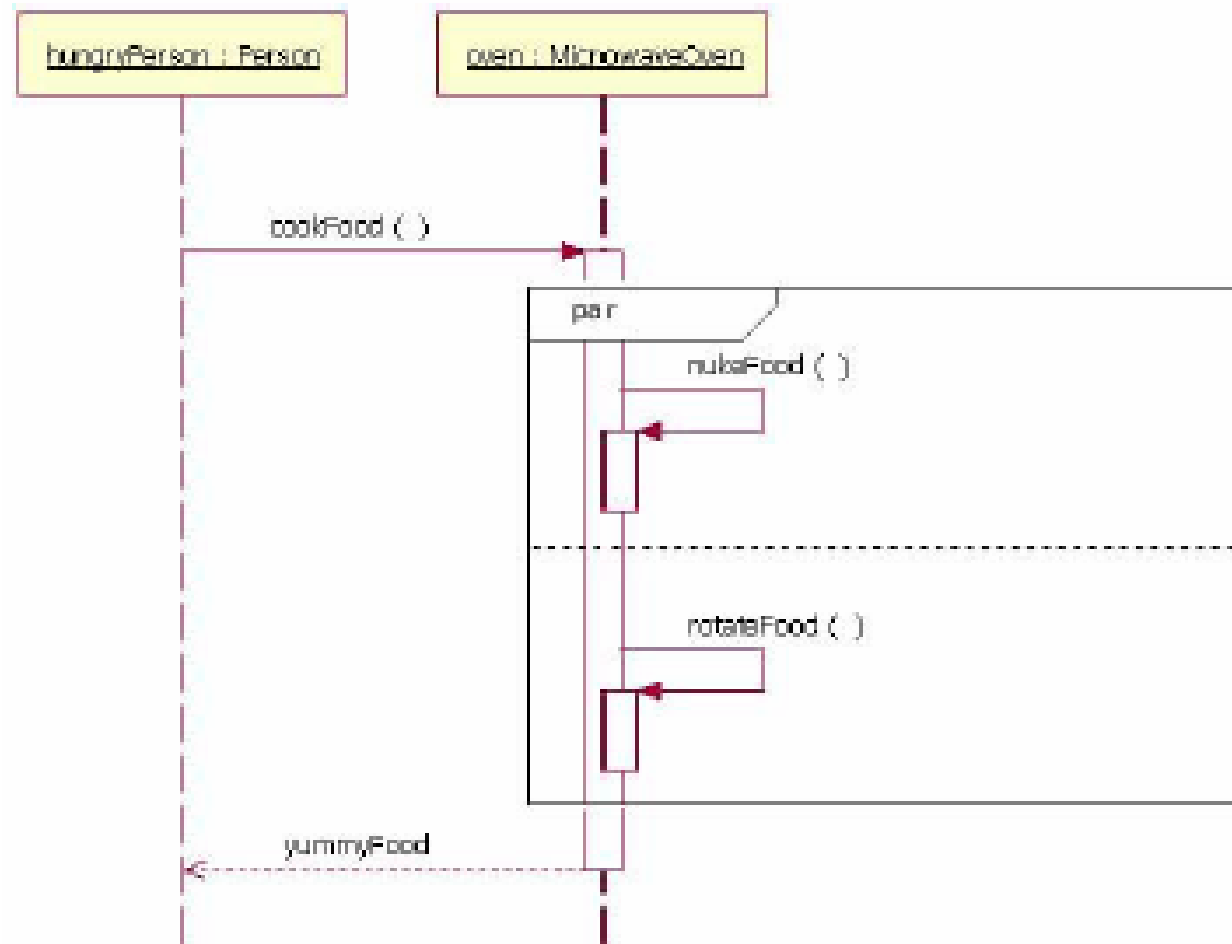




Diagrama de Estado

- Muestra el **ciclo de vida de un objeto**, los **eventos** que recibe, el **estado** entre estos eventos y las **transiciones** de un estado a otro.



Diagrama de Estado (Conceptos)

- **Estado:** Una **condición** o una situación durante la vida de un objeto durante la cual él satisface una cierta condición, realiza una cierta actividad, o espera la ocurrencia de un cierto acontecimiento.
- **Evento:** Una **ocurrencia** de un **acontecimiento significativo**.



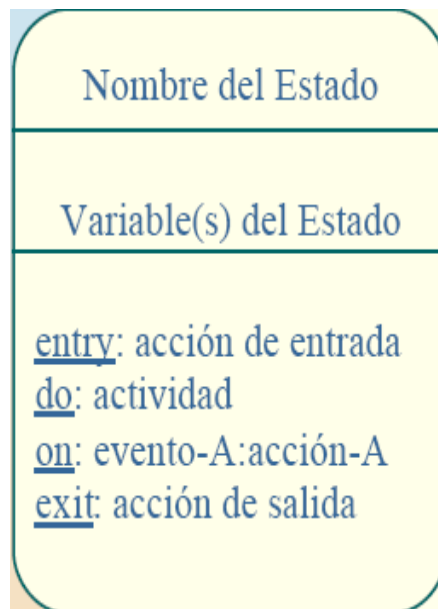
Diagrama de Estado (Conceptos)

- **Transición:** Relación entre dos estados ($e1$ y $e2$) que indica que cuando un evento ocurre, **el objeto pasa del estado $e1$** , en el cual realiza ciertas acciones, **al estado $e2$** cuando un acontecimiento especificado ocurre y condiciones específicas son satisfechas.

Diagrama de Estado y PU

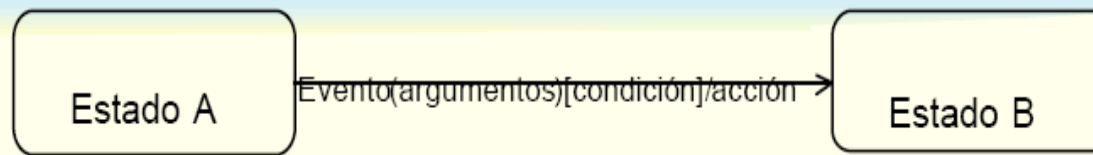
- No existe un modelo llamado « **modelo de estados** », sin embargo cualquier elemento del modelo del dominio o del modelo de diseño puede tener asociado un diagrama de estados para mejorar su comprensión o para modelar su comportamiento dinámico.

Diagrama de Estado (Notación)



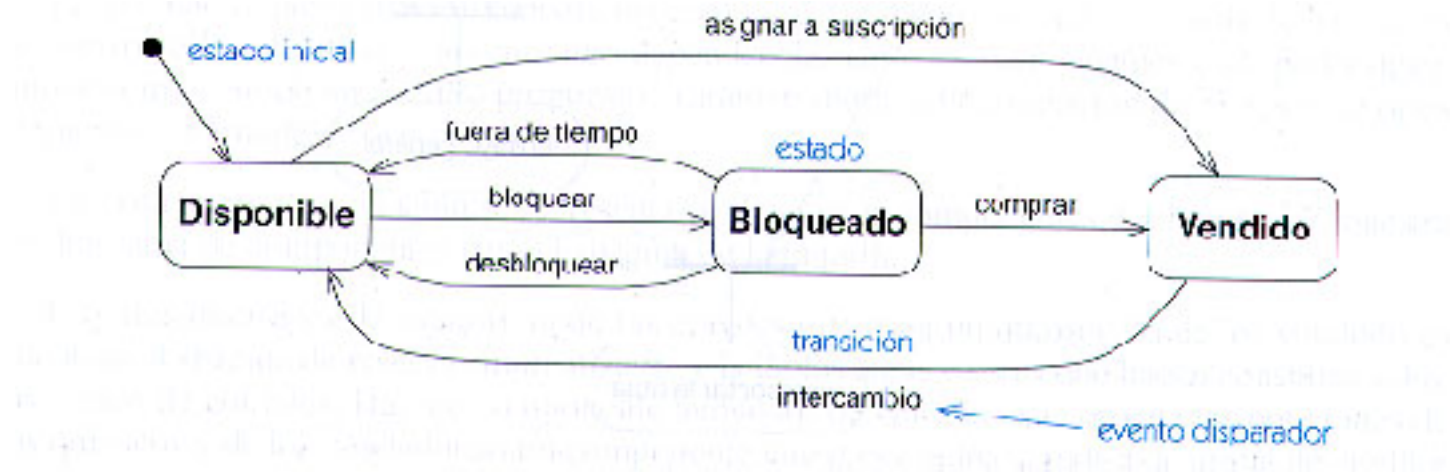
- Acciones Entry/Exit
 - entry: Acción ejecutada en la entrada al estado
 - exit: Acción ejecutada en la salida del estado
- do: Actividad que se está ejecutando mientras permanece en un estado interrumpible
- on: Acción que se ejecuta como resultado de un evento específico

Diagrama de Estado (Notación)



- **Evento**
- **Condición de Guarda**
- **Acción:** Cómputo atómico ejecutable que se dispara como respuesta al evento.

Ejemplo de: Diagrama de Estado



Ejemplo de: Diagrama de Estado y CU

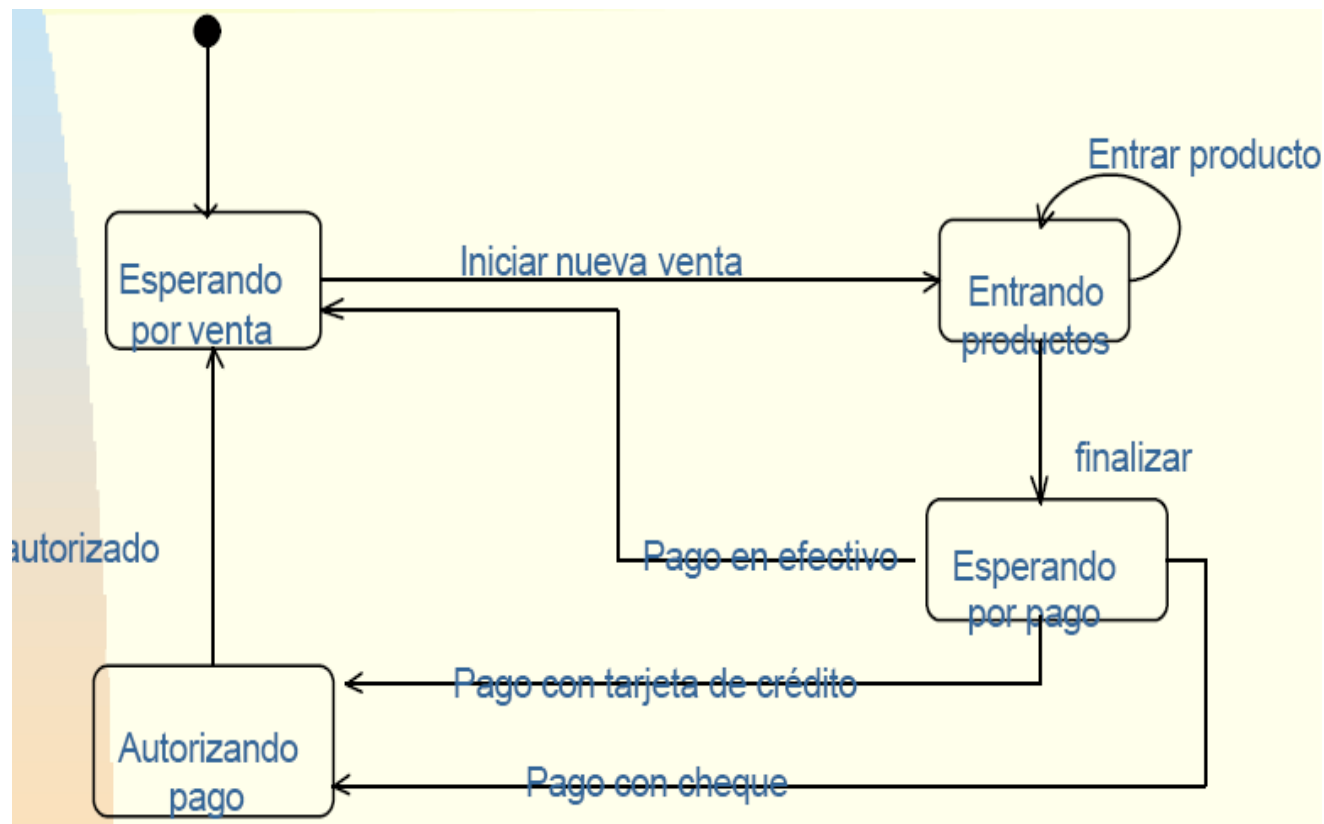
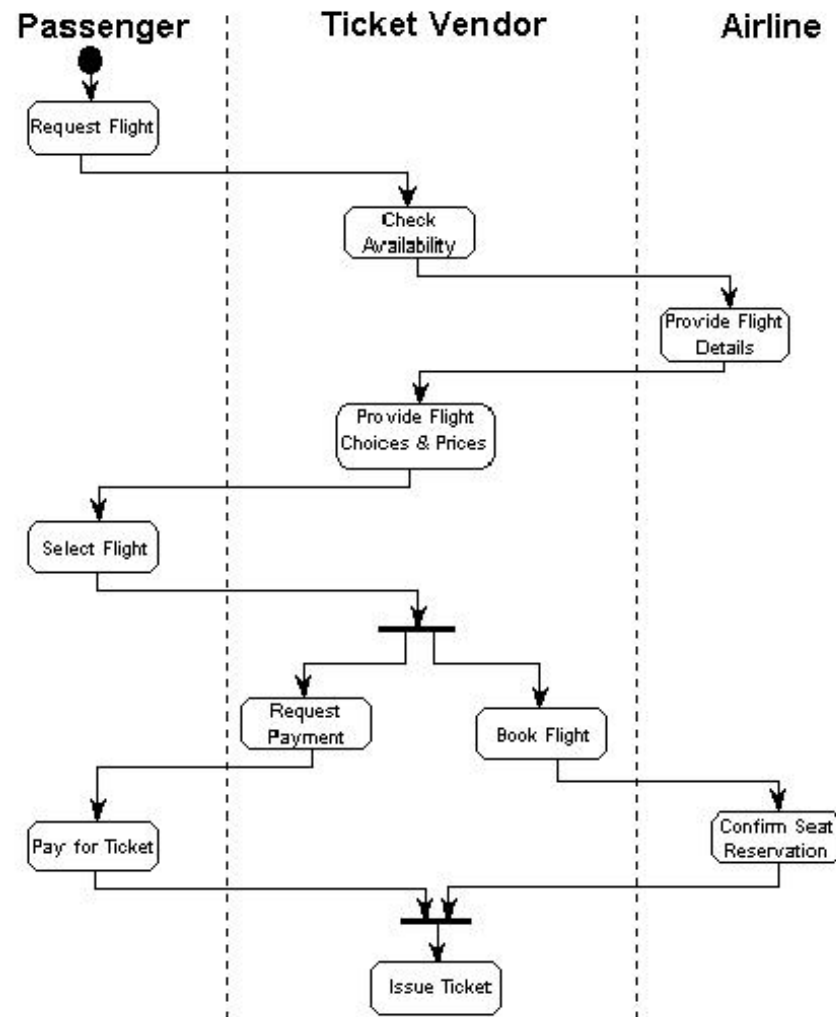


Diagrama de Actividad

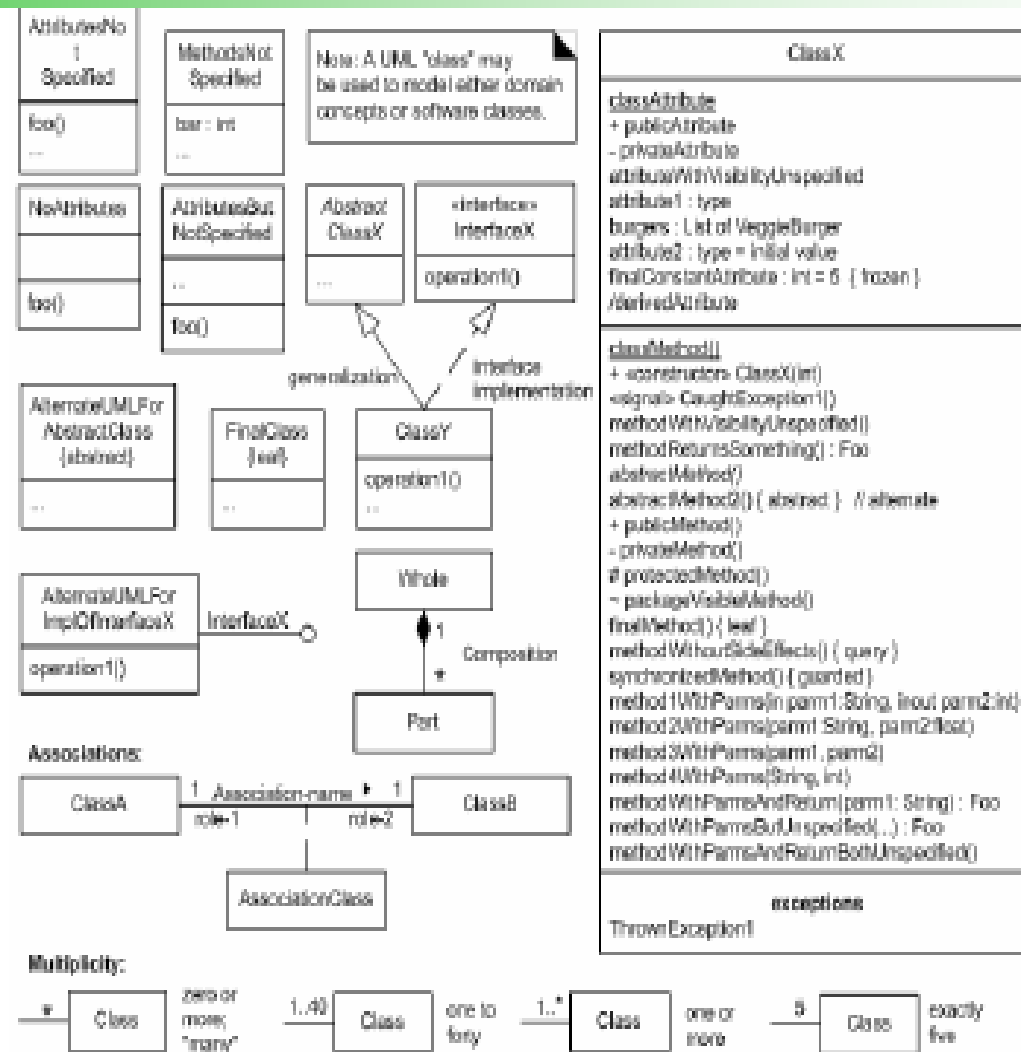
- El Diagrama de Actividad es una especialización del Diagrama de Estado, organizado respecto de las acciones y usado para especificar:
 - Un método
 - Un caso de uso
 - Un proceso de negocio (Workflow)

Ejemplo de: Diagrama de Actividad



Ejemplo de:

Diagrama de Clases



Ejemplo de: Estructuración en Paquetes

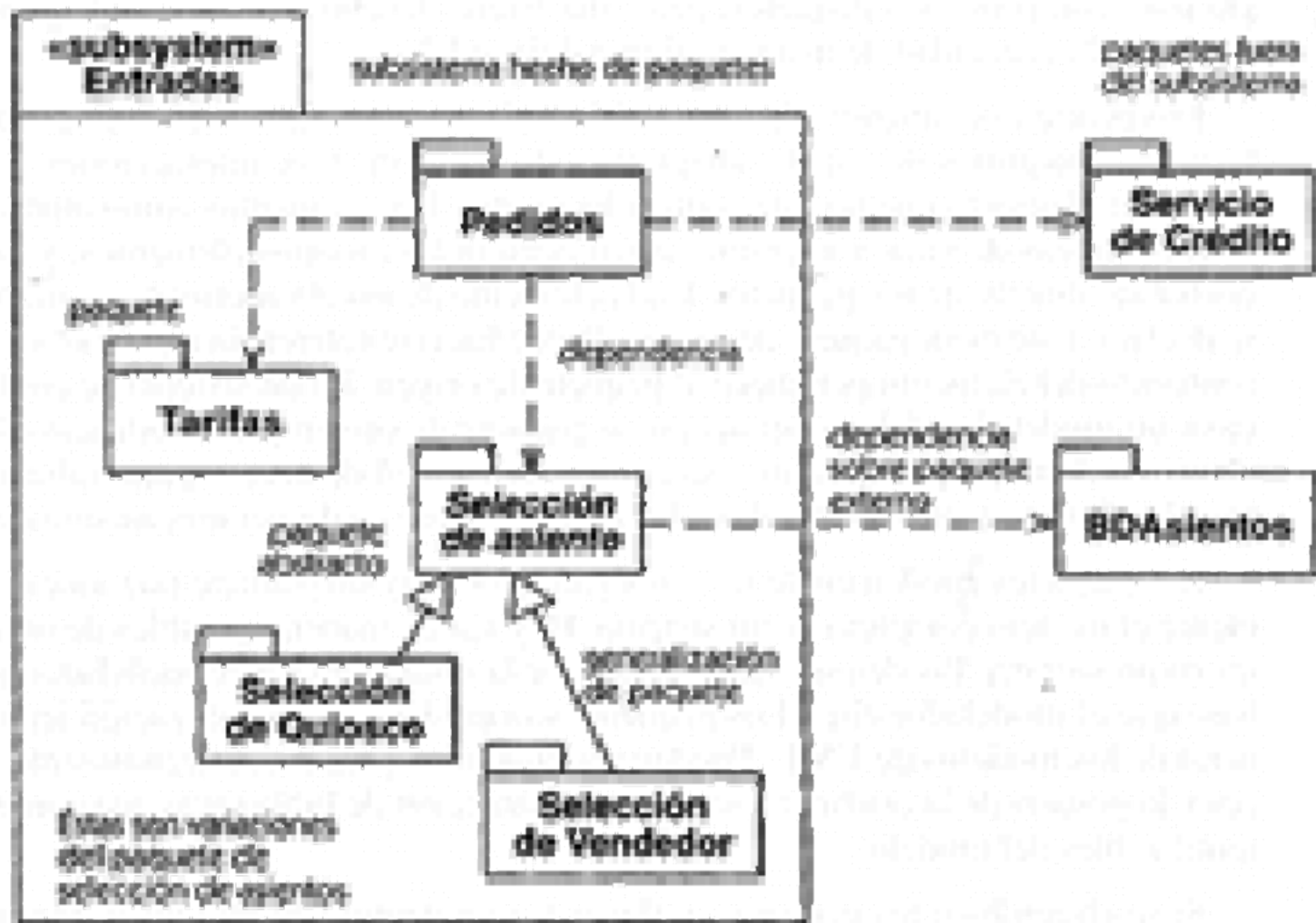




Diagrama de Componentes

- Un **diagrama de componentes** muestra las **dependencias lógicas** entre **componentes** software.
- Un **componente** representa una parte de un sistema modular, desplegable y reemplazable, que encapsula la implementación y expone un conjunto de interfaces.
- Ejemplo: Código fuente, binario o ejecutable.

Diagrama de Componentes

- La dependencia entre dos componentes se muestra como una flecha punteada.
- La dependencia quiere decir que una componente necesita de la otra para completar su definición

Ejemplo de Diagrama de Componentes (Vista de Implementación)

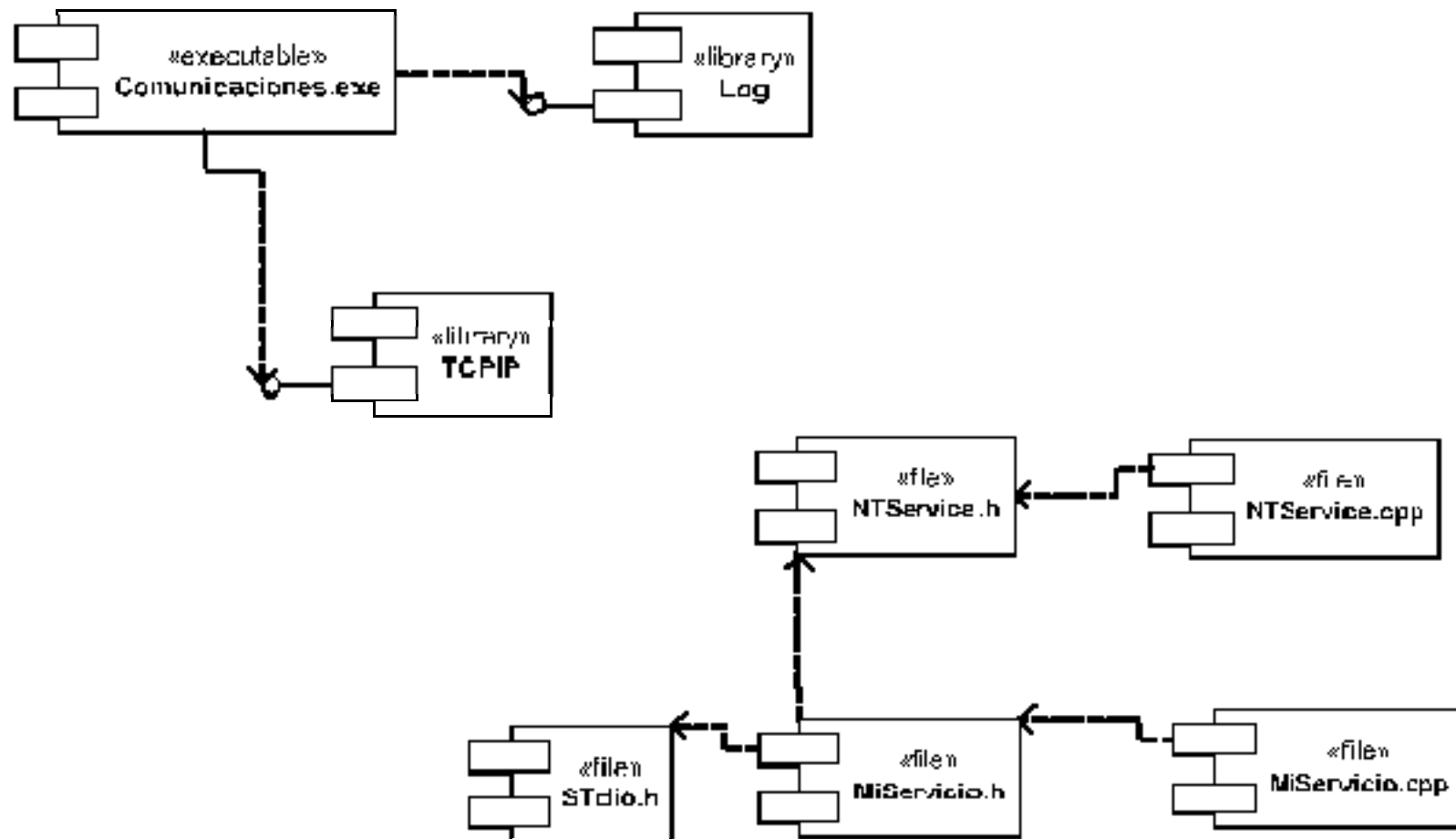




Diagrama de Despliegue

- Muestra cómo se configuran las instancias de los componentes y los procesos para la ejecución run-time en las instancias de los nodos de proceso
- Describen la arquitectura física del sistema durante la ejecución, en términos de:
 - procesadores
 - dispositivos
 - componentes de software

Diagrama de Despliegue

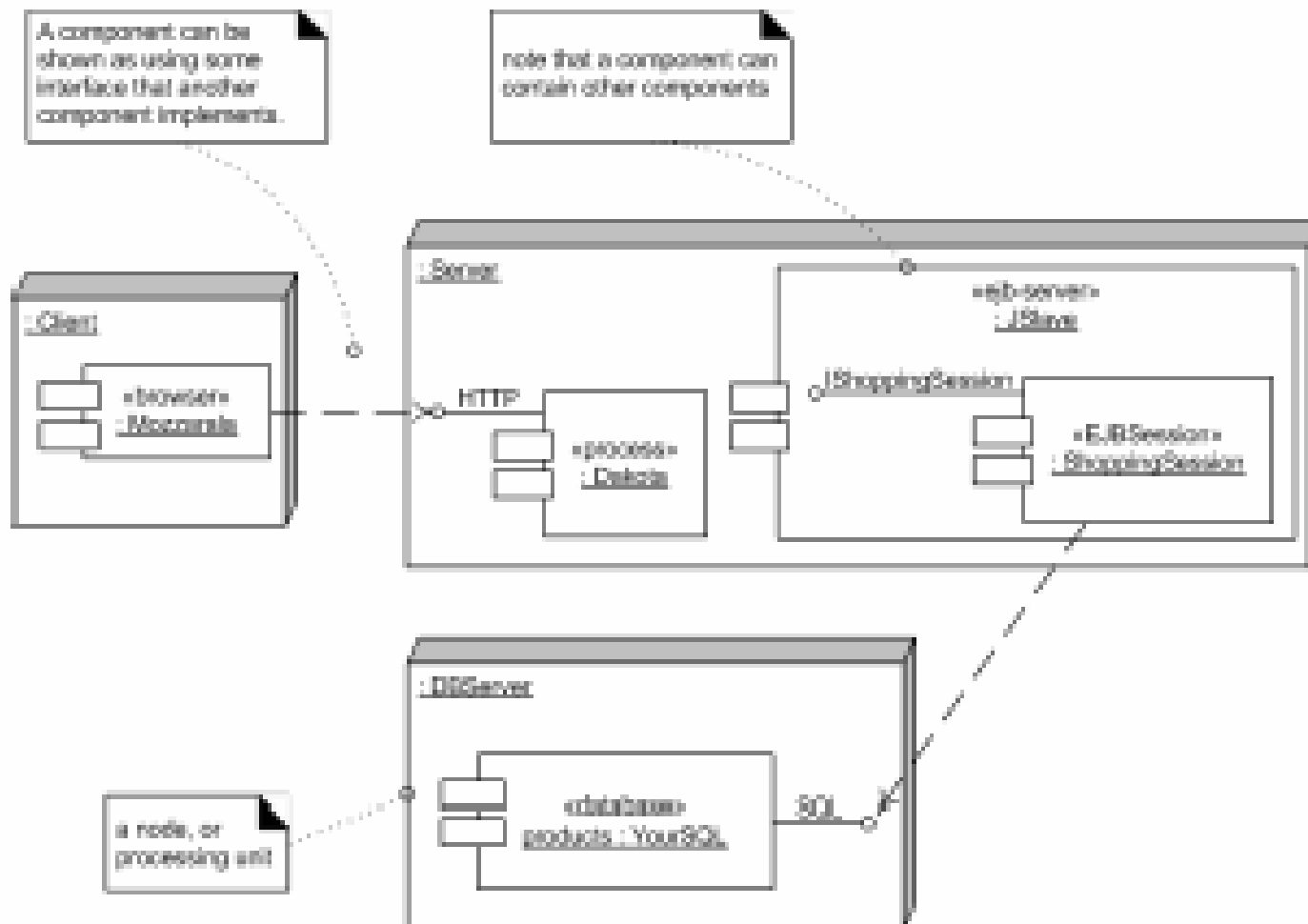
- Un nodo es un objeto físico en tiempo de ejecución que representa un recurso computacional, generalmente con memoria y capacidad de procesamiento.
- Los dispositivos del sistema también se representan como nodos.
- Generalmente se usan estereotipos para identificar el tipo de dispositivo.



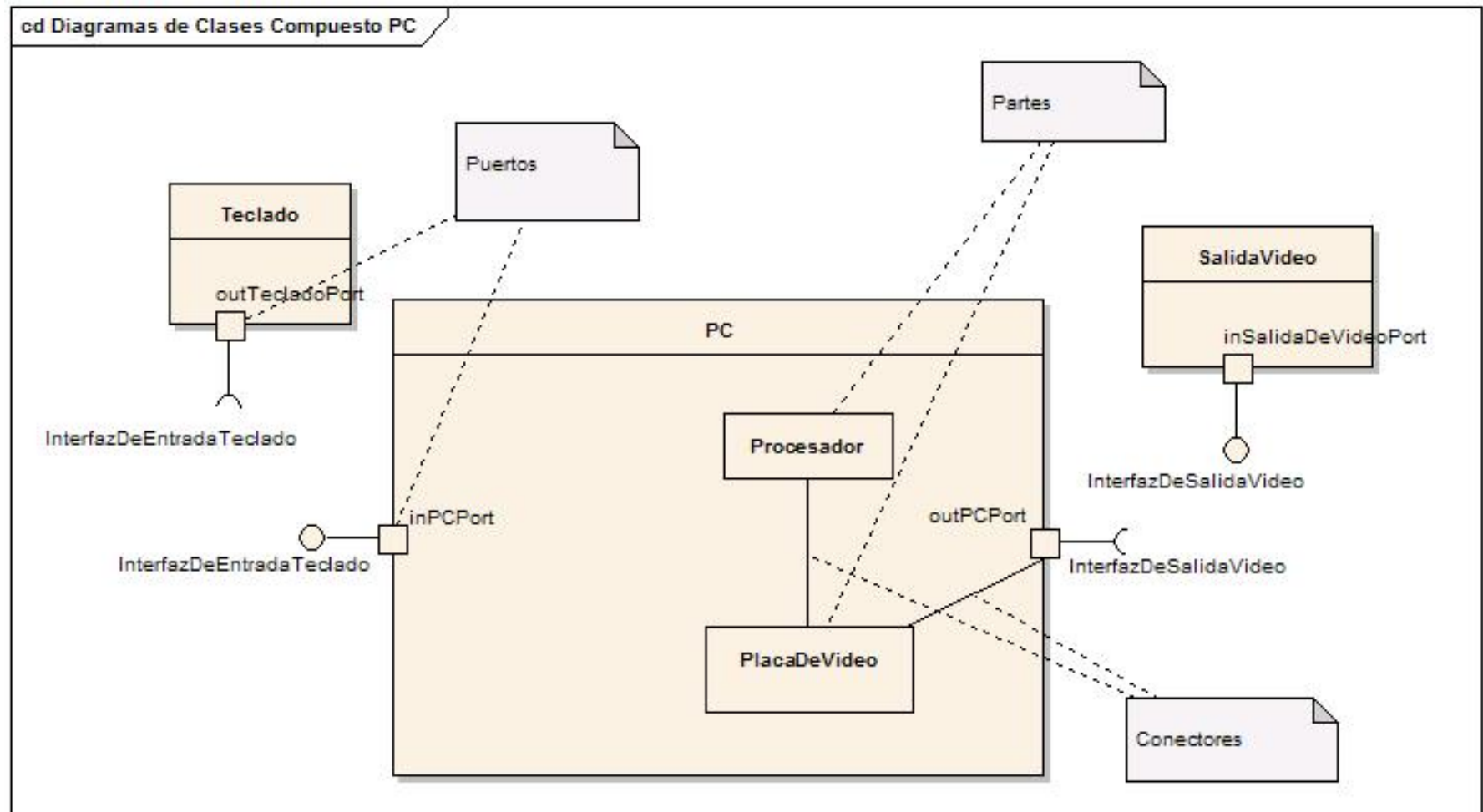
Diagrama de Despliegue

- Los nodos se conectan mediante asociaciones de comunicación.
- Estas asociaciones indican:
 - Algún tipo de ruta de comunicación entre los nodos
 - Los nodos intercambian objetos o envían mensajes a través de esta ruta
- El tipo de comunicación se identifica con un estereotipo que indica el protocolo de comunicación o la red.

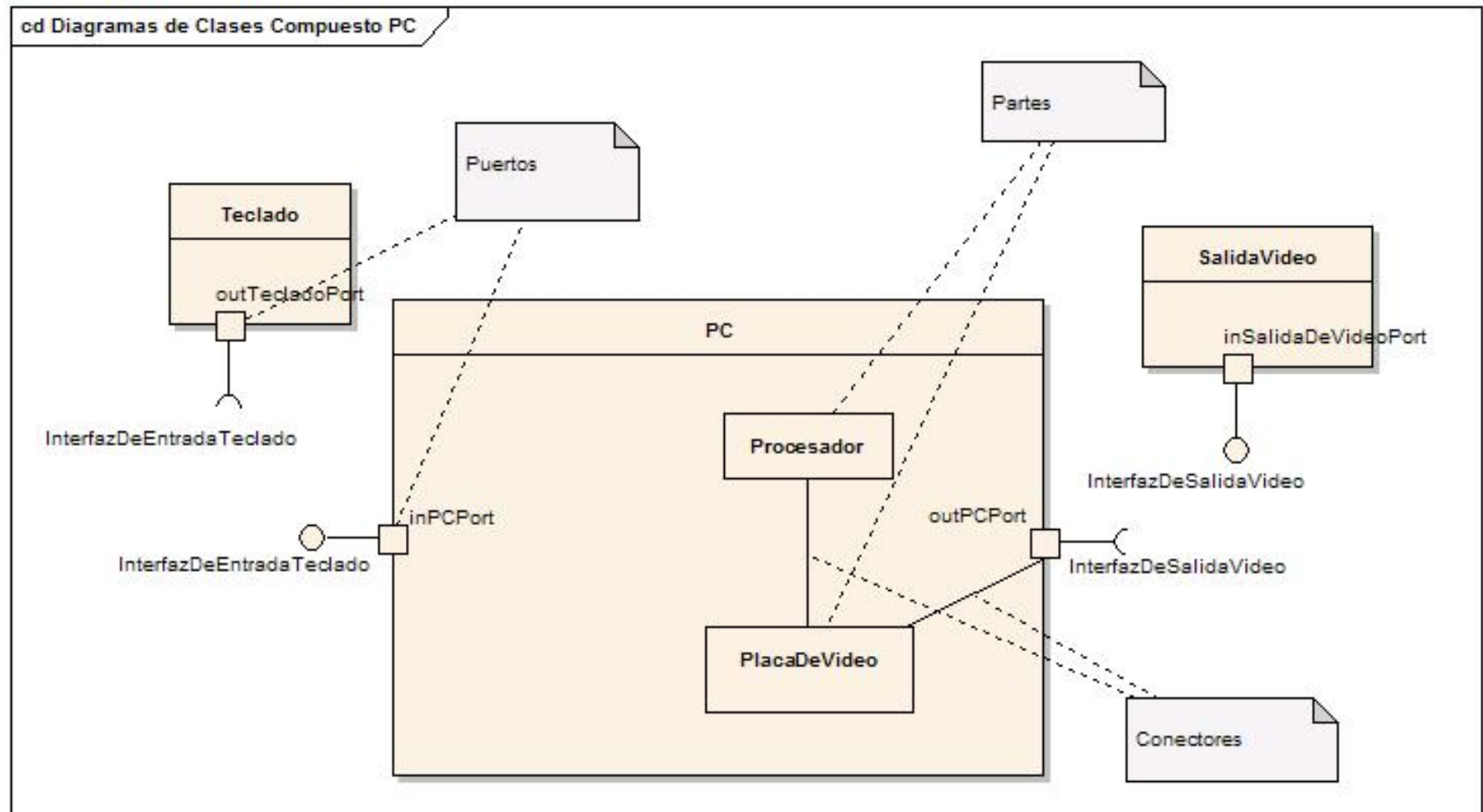
Ejemplo de: Diagrama de Despliegue



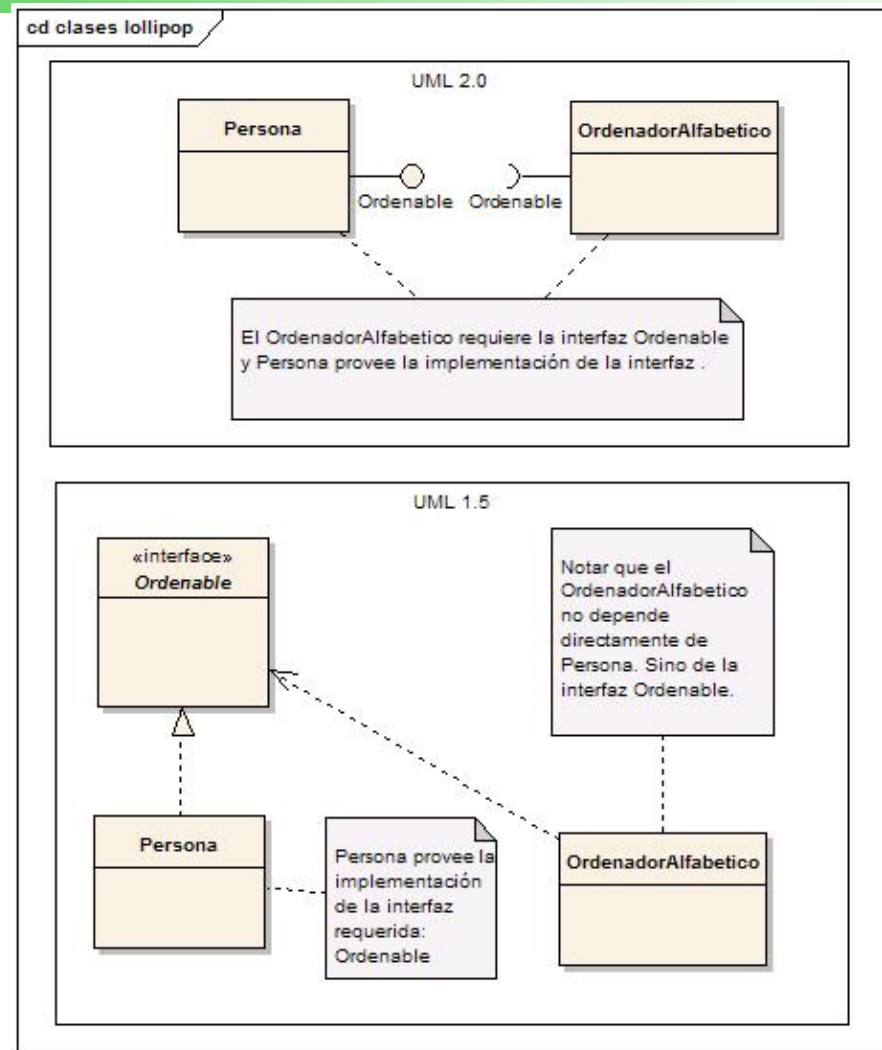
Nuevas Características de UML 2.0



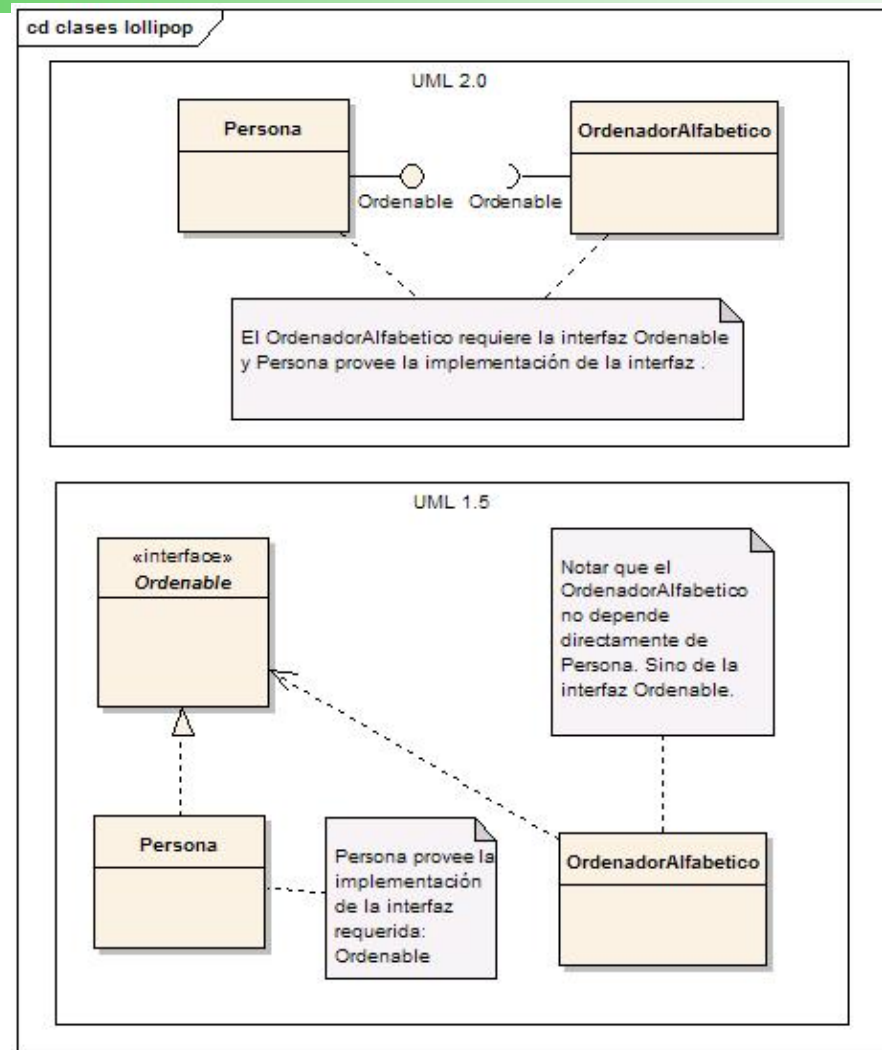
Nuevas Características de UML 2.0



Nuevas Características de UML 2.0



Nuevas Características de UML 2.0



Nuevas Características de UML 2.0

