



INGENIERÍA DEL SOFTWARE I

Tema 9

Interacciones del Sistema (en desarrollo OO)

Univ. Cantabria – Fac. de Ciencias
Francisco Ruiz



Objetivos del Tema

- Conocer en detalle los conceptos de **Interacciones**.
- Aprender a realizar **diagramas de secuencia y de comunicación** de UML 2.
- Aprender a **modelar** con ellos **flujos de control**
 - Por ordenación temporal de Mensajes, o
 - Por organización de objetos.
- Comprender los **usos diferentes** de los diagramas de interacción durante los **requisitos** y durante el **diseño**.



Contenido

- Introducción
- Conceptos de Interacciones
 - Interacción
 - Objeto y Rol
 - Enlace y Conector
 - Mensaje
 - Secuenciación
 - Creación, Modificación y Destrucción de Objetos
- Diagramas de Interacción
 - De Secuencia
 - De Comunicación
 - Control Estructurado
 - Consejos
- Modelado
 - Flujo de Control por Tiempos
 - Flujo de Control por Organización
 - Requisitos vs Diseño



Bibliografía

- Básica
 - Booch, Rumbaugh y Jacobson (2006): El Lenguaje Unificado de Modelado. 2ª edición.
 - Caps. 16 y 19.
- Complementaria
 - Rumbaugh, Jacobson y Booch (2007): El Lenguaje Unificado de Modelado. Manual de Referencia. 2ª edición.
 - Cap. 9.
 - Booch, Rumbaugh y Jacobson (2006): El Lenguaje Unificado de Modelado. 2ª edición.
 - Caps. 23 y 24.



Introducción

- En cualquier **sistema OO** los **objetos** interactúan entre sí pasándose **mensajes**.
- Una **interacción** es un comportamiento que incluye un conjunto de mensajes que se intercambian un conjunto de objetos dentro de un contexto para lograr un propósito.
- Las interacciones se utilizan para modelar los **aspectos dinámicos de las colaboraciones**.



Introducción

- **Colaboración**
 - Sociedad de objetos que desempeñan roles específicos y colaboran entre sí para desarrollar un comportamiento mayor que la suma de los comportamientos de sus elementos.
 - Estos roles representan instancias típicas de clases, interfaces, componentes, nodos y casos de uso.
 - Los aspectos dinámicos se modelan como flujos de control
 - Con hilos secuenciales sencillos a través de un sistema, o
 - Flujos más complejos con bifurcaciones, iteraciones, recursión y concurrencia.



Conceptos de Interacciones

- Al trabajar con interacciones se manejan, entre otros, los siguientes conceptos:
 - Interacción
 - Objeto
 - Rol
 - Enlace
 - Conector
 - Mensaje
 - Secuenciación



Conceptos - Interacción

- Definición de **Interacción**

Una comportamiento dado por un conjunto de mensajes que se intercambian entre un conjunto de objetos dentro de un contexto para lograr un propósito.



Conceptos - Interacción

- En UML los **aspectos dinámicos** de los sistemas se modelan mediante **interacciones**
 - Porque los objetos interactúan para realizar colectivamente los servicios ofrecidos por las aplicaciones.
- Una interacción establece el escenario presentando:
 - Todos los objetos que colaboran (similar a diagramas de objetos), y
 - Los mensajes enviados entre los objetos (diferente a diagramas de objetos).
- Las interacciones se utilizan para **modelar el flujo de control** dentro de
 - una operación, clase, componente, caso de uso, subsistema o sistema completo.



Conceptos - Interacción

- Las interacciones aparecen en la colaboración de objetos existentes en el **contexto de**
 - Un **sistema o subsistema**
 - Pueden **colaborar** objetos locales o de **distintas partes** del sistema o con distintos niveles conceptuales.
 - Una **operación**
 - Los parámetros, variables locales y objetos globales a la operación (visibles por ella) pueden interactuar entre sí para llevar a cabo el **algoritmo que la implementa**.
 - Un **clasificador**
 - Las interacciones sirven para visualizar, especificar, construir y documentar la **semántica de** un clasificador (**clase, componente, nodo o caso de uso**).
 - En el contexto de un caso de uso la interacción representa un escenario (flujo particular).



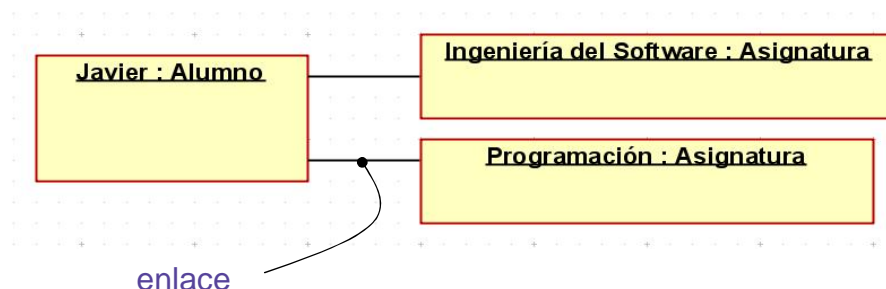
Conceptos - Objeto y Rol

- Los **objetos** que participan en una interacción son
 - Elementos **concretos**: (**INSTANCIAS**)
 - Representan algo del mundo real.
 - p, instancia de la clase Persona, representa a una persona particular.
 - Elementos **prototípicos**: (**ROLES**)
 - Representan cualquier elemento de un cierto tipo.
 - P representa cualquier instancia de la clase Persona, es decir, cualquier persona.
- En el contexto de una interacción se pueden encontrar **instancias de**:
 - Clases, componentes, nodos y casos de uso.
 - Las instancias de **clases abstractas** y de **interfaces** en realidad representan a instancias indirectas (prototípicas o roles) de cualquier clase, hija de la clase abstracta o que realiza la interfaz.



Conceptos – Enlace y Conector

- **Enlaces**:
 - Conexiones semánticas entre objetos.
 - Habitualmente son instancias de asociación.
 - Especifican un camino a lo largo del cual un objeto puede enviar un mensaje a otro objeto (o a sí mismo).





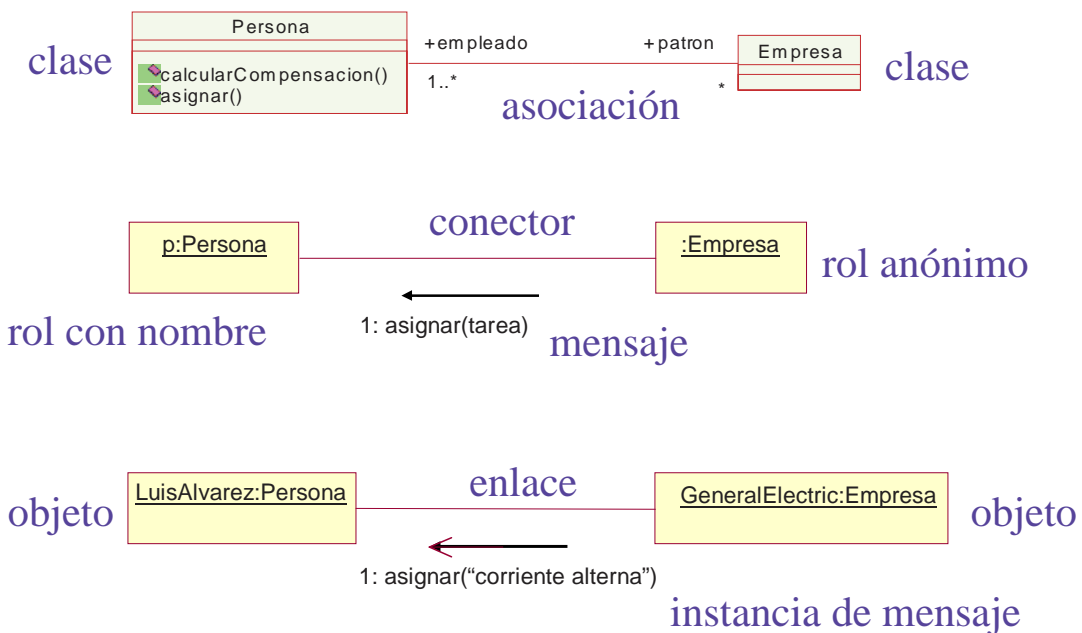
Conceptos - Enlace y Conector

- Si una clase tiene una asociación con otra clase =>
 - Puede existir un enlace entre instancias de las dos clases.
- Si hay un enlace entre dos objetos =>
 - Un objeto puede enviar un mensaje al otro.
- Igual que los objetos prototípicos se llaman roles, los **enlaces prototípicos** se llaman **conectores**.



Conceptos - Enlace y Conector

- Ejemplo de **Asociación, Conector y Enlace**.





Conceptos - Enlace y Conector

- Existen **estereotipos** para adornar el extremo del enlace indicando la naturaleza del enlace (motivo de visibilidad del objeto):
 - **association** existe una asociación
 - **self** es el invocador de la operación
 - **global** su ámbito contiene el actual
 - **local** su ámbito es local
 - **parameter** es un parámetro



Conceptos – Mensaje

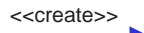
- **Mensajes:**
 - Especificación de una **comunicación entre objetos** que **transmite información** con la expectativa de que se desencadenará una actividad.
 - La **recepción** de una instancia de un mensaje es una ocurrencia de un evento.
 - Cuando se pasa un mensaje, su recepción suele producir una **acción**.
 - La acción puede producir cambios en el estado del objeto destinatario y en los objetos accesibles desde él.



Conceptos – Mensaje

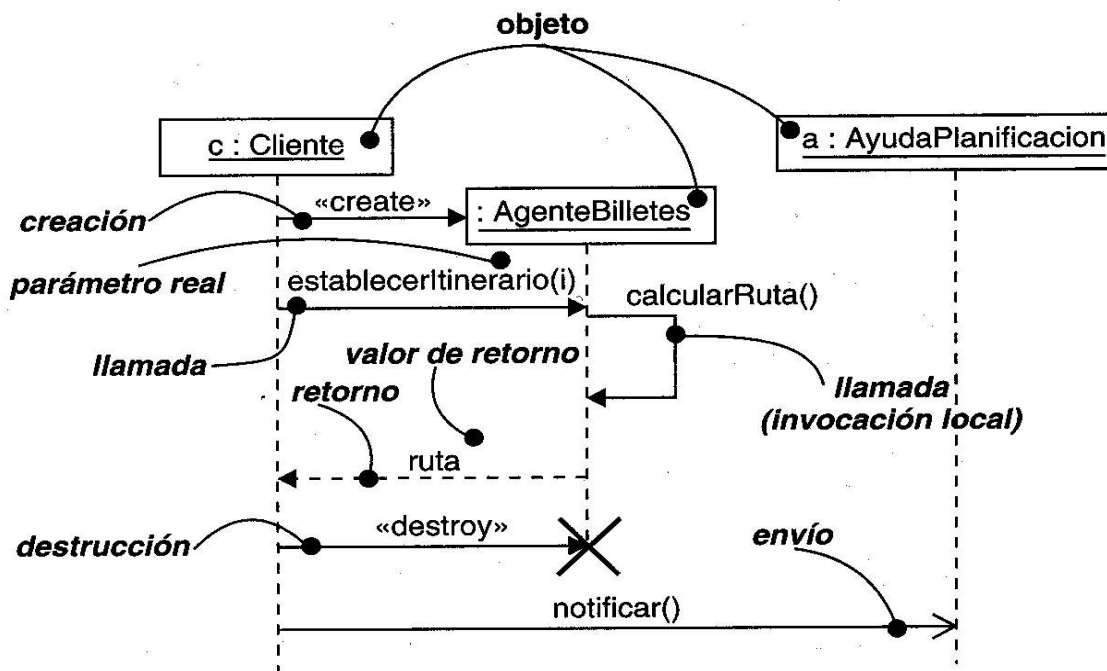
• Tipos de Acciones en Mensajes:

- **Llamada:** Invoca una **operación**.
 - Mensaje **síncrono**: el emisor espera hasta recibir el resultado.
 - **Envío:** Envía una **señal** (valor objeto).
 - Mensaje **asíncrono**: el emisor no espera a recibir el resultado.
 - **Retorno:** Devuelve un valor al emisor.
 - **Creación:** Crea un objeto.
 - **Destrucción:** Destruye un objeto. Un objeto puede "suicidarse" al destruirse a sí mismo.
-
- Las llamadas y envíos pueden incluir **parámetros**.
 - Los retornos pueden tener un **valor de retorno**.



Conceptos – Mensaje

• Tipos de Acciones en Mensajes:





Conceptos – Mensaje

- **Sintaxis** para mensajes:

$\{<predecesor>|<guarda>\} <secuencia> : <retorno> := <nombre_mensaje>(<argumentos>)$

- **<predecesor>**: lista separada por comas de los números de secuencia de mensajes que deben ocurrir antes del mensaje especificado.
- **<guarda>**: condición para el envío del mensaje.
- **<secuencia>**: representa el nivel de anidamiento procedural.
 - 3.1.4 es posterior al 3.1.3 dentro de la activación 3.1.
 - Se pueden añadir nombres para especificar **mensajes concurrentes**: los mensajes 3.1a y 3.1b son concurrentes dentro de la activación 3.1.
 - Se puede incluir una especificación de **iteración**:
 - $*[i:=1..n]$ => envío de una **secuencia** de mensajes
 - $*||[i:=1..n]$ => el envío es en **paralelo**.
- Solo el nombre del mensaje es obligatorio.



Conceptos – Mensaje

- **Ejemplos** de Mensajes:

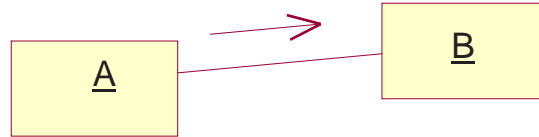
- Mensaje Simple:
 - 2: Visualizar (x,y)
- Llamada Anidada con valor de retorno:
 - 1.3.1: p := buscar(especificaciones)
- Mensaje Condicional:
 - [x < 0] 4: invertir(x,color)
- Iteración secuencial:
 - 3.1 *: modificar()
- Iteración en paralelo de 3 mensajes:
 - 4: *||[i=1..3] : comunicar
- Sincronización con otros hilos de ejecución:
 - A3, B4 | C3.1 : actualizar



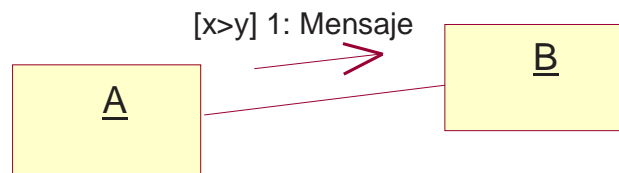
Conceptos – Mensaje

• Ejemplos de Mensajes:

- Un mensaje se envía si han sido enviados los mensajes de una lista (sincronización):



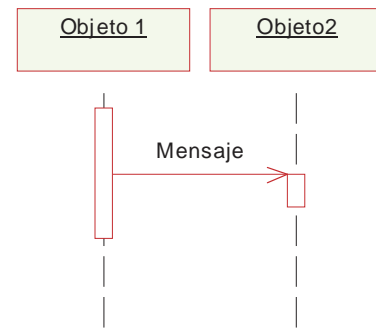
- Un mensaje solo se envía si se cumple una condición:



Conceptos - Secuenciación

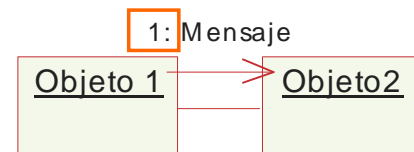
• Una **secuenciación** es una **secuencia de mensajes entre objetos**.

- Cada proceso e hilo de un sistema define un flujo de control separado, y dentro de cada flujo los mensajes se ordenan en secuencia temporal.



- Para mejorar la visualización:

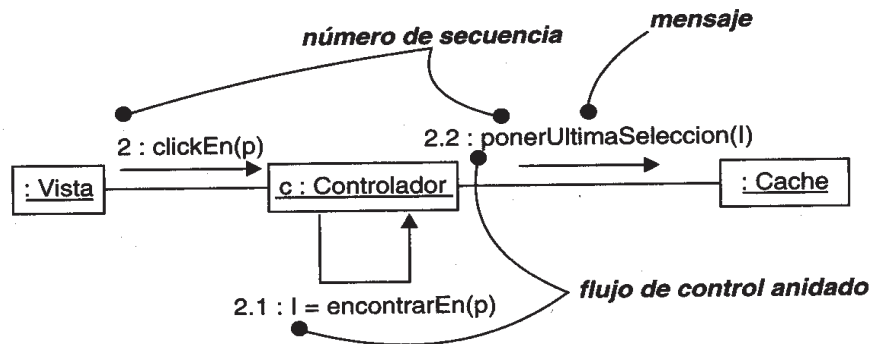
- Expresar la posición de un mensaje con relación al inicio de la secuencia
 - <nº secuencia>: <mensaje>





Conceptos - Secuenciación

- Tipos de flujo de control en una **secuenciación**:
 - Procedimental** o Anidado → **Síncrono**



- Plano** (sin anidamiento) → **Asíncrono**



Conceptos - Secuenciación

- Cuando hay **varios flujos de control**
 - Es importante identificar el proceso o hilo que envía cada mensaje.
 - Para ello se utiliza el predecesor:
 - D5: abrirEscotilla()** =>
 - La operación **abrirEscotilla** se envía como el quinto mensaje de la secuencia cuya raíz es el proceso o hilo identificado como D.



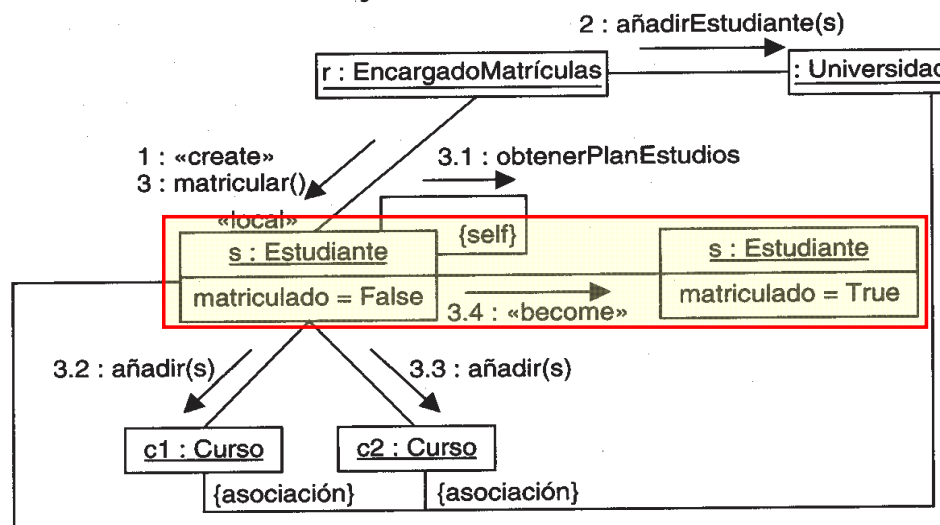
Conceptos - Creación, Modificación y Destrucción

- Normalmente los objetos existen durante todo el tiempo que dura la interacción.
- Pero a veces los mensajes conllevan crear (*create*) o destruir (*destroy*) objetos (o enlaces) durante una interacción.
- Se indica mediante **restricciones**:
 - **Creación (*new*)** → La instancia/enlace se crea durante la ejecución de la interacción que lo contiene.
 - **Destrucción (*destroyed*)** → La instancia/enlace se destruye antes de acabar la interacción que lo contiene
 - **Modificación (*transient*)** → La instancia/enlace se crea durante la ejecución de la interacción y se destruye antes de completarse la misma.



Conceptos - Creación, Modificación y Destrucción

- A lo largo de una interacción, **un objeto suele cambiar** los valores de sus atributos, estado o roles.
 - Se puede representar incluyendo *varias copias del mismo objeto* conectadas con un mensaje *<<become>>*.





Diagramas de Interacción

- **RECORDATORIO**

- Son un grupo especial de diagramas de comportamiento que muestran una **interacción**:
 - Conjunto de objetos o roles y mensajes que pueden ser enviados entre ellos.
- Los diagramas de interacción muestran cómo se comunican los objetos en una interacción.
- UML 2 incluye los siguientes
 - **Secuencia**
 - **Comunicación** (antiguo de Colaboración en UML 1.x)
 - **Tiempos**
 - **Revisión de las Interacciones**



Diagramas de Interacción

- Se pueden **usar para**:
 - Modelar un **flujo de control de un caso de uso** (dominio del problema).
 - Visualizar, especificar, construir y documentar la **dinámica** de una **sociedad de objetos** (dominio de la solución).
- Paralelismo con una película
 - Cuando vemos una película nuestra mente nos engaña. No existe movimiento continuo, sino una serie de **imágenes estáticas** reproducidas tan rápidamente que producen la ilusión del movimiento continuo.
 - **Los directores planifican** la película utilizando **viñetas** gráficas con los fotogramas clave de la película.
 - Modelo de cada **escena** con el detalle suficiente para transmitir su intención a todos los integrantes del equipo de producción.



Diagramas de Interacción

- **Contenido:**

- Objetos
- Enlaces
- Mensajes
- Opcionalmente pueden incluir anotaciones (comentarios y restricciones)

- Los objetos pueden ser **instancias de:**

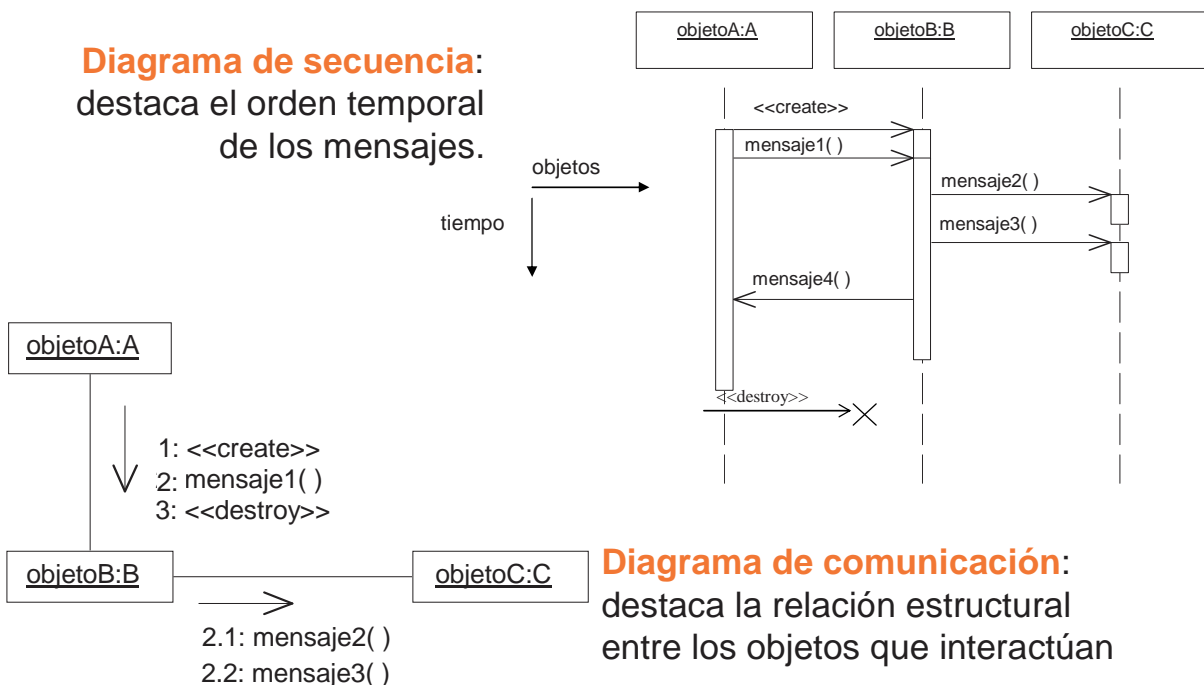
- Clases
- Interfaces
- Componentes
- Nodos



Diagramas de Interacción

RECORDATORIO

Diagrama de secuencia:
destaca el orden temporal de los mensajes.

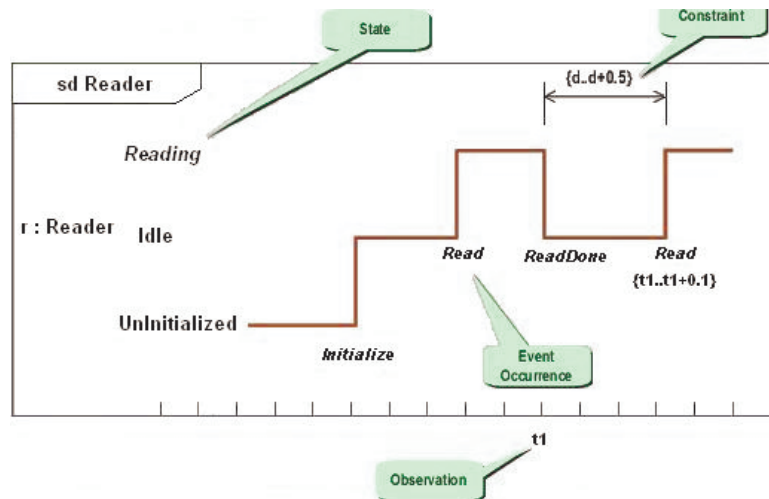




Diagramas de Interacción

De Tiempos: RECORDATORIO

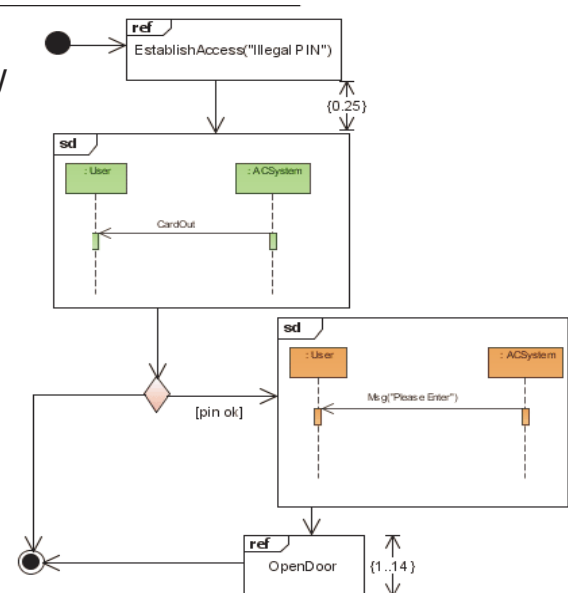
- Muestran los tiempos reales entre diferentes objetos o roles.
 - Comportamiento de los objetos en un periodo determinado de tiempo.
- Son una forma especial de diagramas de secuencia (los ejes están girados).



Diagramas de Interacción

De Revisión de Interacciones: RECORDATORIO

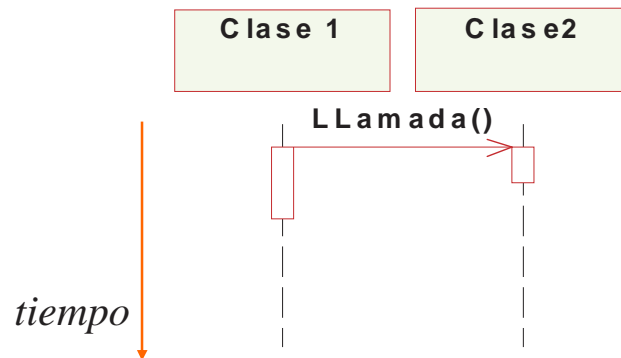
- Aportan una **visión general** del flujo de control de las interacciones.
- Híbrido entre diagrama de actividad y diagrama de secuencia.
- También llamados
Visión Global de Interacciones





Diagramas de Interacción – De Secuencia

- Un **Diagrama de Secuencia** es un diagrama de interacción que resalta la ordenación temporal de los mensajes.
- Presentan un conjunto de **roles** y los **mensajes** enviados y recibidos por las instancias que interpretan dichos roles.
- Tiene forma de tabla con objetos dispuestos en horizontal y mensajes en vertical, ordenados temporalmente.

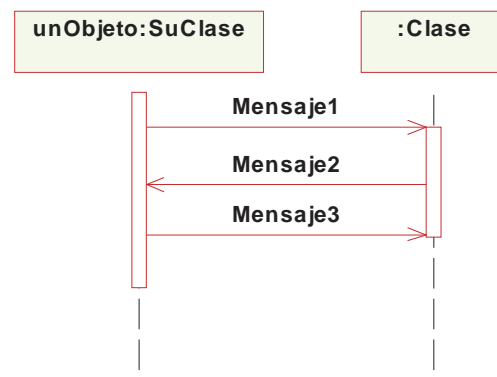


Diagramas de Interacción – De Secuencia

- Un diagrama de secuencia se forma:

1. Colocando los **objetos** que participan en la interacción en la parte superior del diagrama (**eje X**).

- Normalmente a la izquierda el objeto que inicia la interacción, y los objetos subordinados a la derecha.

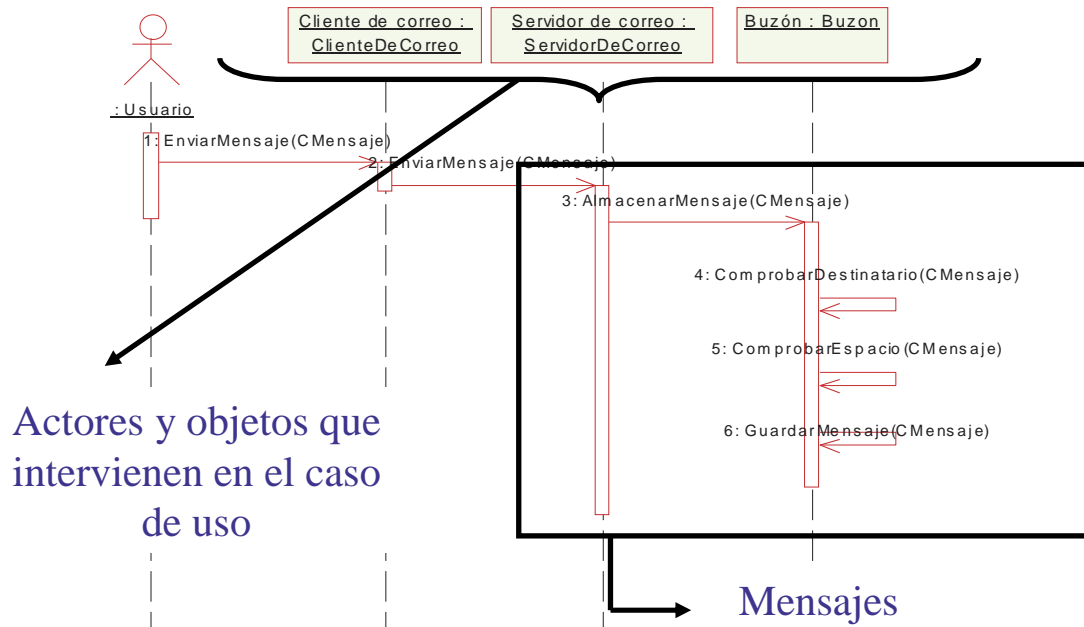


2. A continuación, se colocan los **mensajes** que estos objetos envían y reciben a lo largo del eje Y, en orden de sucesión en el tiempo, desde arriba hasta abajo.



Diagramas de Interacción – De Secuencia

Escenario normal (diagrama de secuencia): el mensaje llega a su destino.

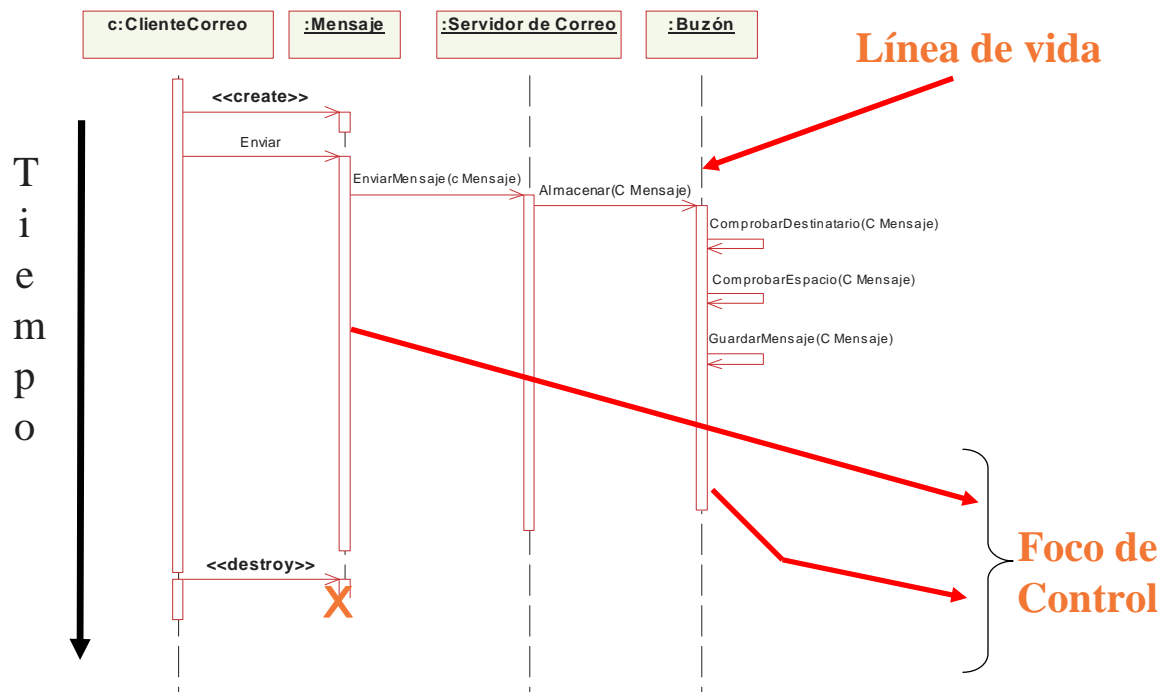


Diagramas de Interacción – De Secuencia

- Tienen dos características que los distinguen de los diagramas de comunicación:
 - La **Línea de Vida**
 - Representa la existencia de un objeto a lo largo de un período de tiempo.
 - Puede tener **bifurcaciones**, que después deben unirse.
 - El **Foco de Control**
 - Representa el período de tiempo durante el cual un objeto ejecuta una acción.
 - Se muestra como un rectángulo en la línea de vida.
 - La parte superior se alinea con el comienzo de la acción.
 - La parte inferior se alinea con su terminación. Puede tener un mensaje de retorno.



Diagramas de Interacción – De Secuencia



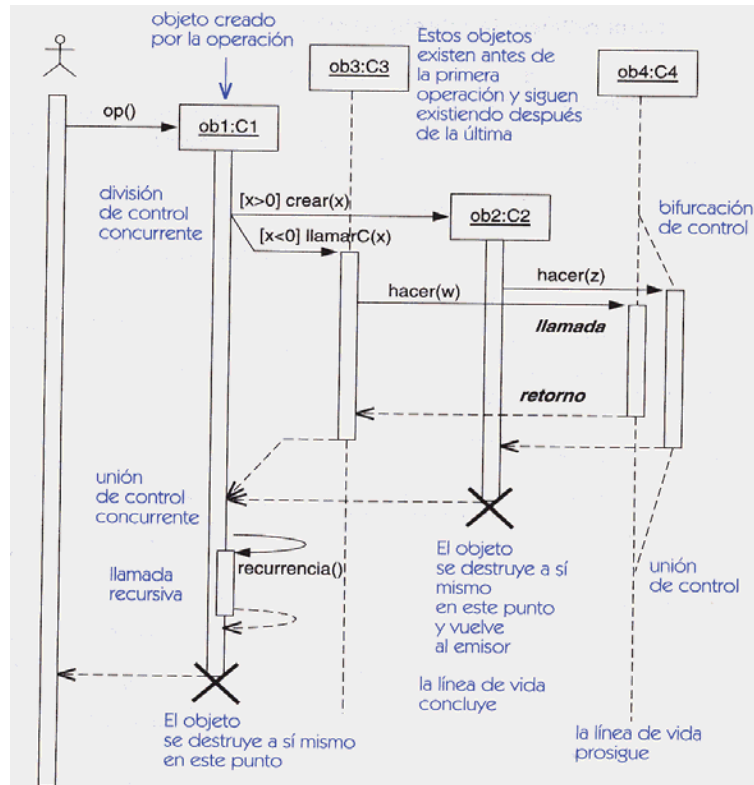
Diagramas de Interacción – De Secuencia

- Con la línea de vida se puede indicar la **creación o destrucción de objetos** o roles durante la interacción:
 - Objetos **creados**
 - Su línea de vida comienza con un mensaje <<create>> que va hasta el objeto.
 - Objetos **destruidos**
 - Su línea de vida acaba con la recepción del mensaje <<destroy>> y se añade el símbolo.
- Con el foco de control se puede mostrar el **anidamiento de llamadas** (llamar a una operación propia):
 - Colocar otro foco de control ligeramente a la derecha del foco padre, medio superpuesto.
 - La llamada puede ser recursiva.



Diagramas de Interacción – De Secuencia

• Ejemplo:



Diagramas de Interacción – De Secuencia

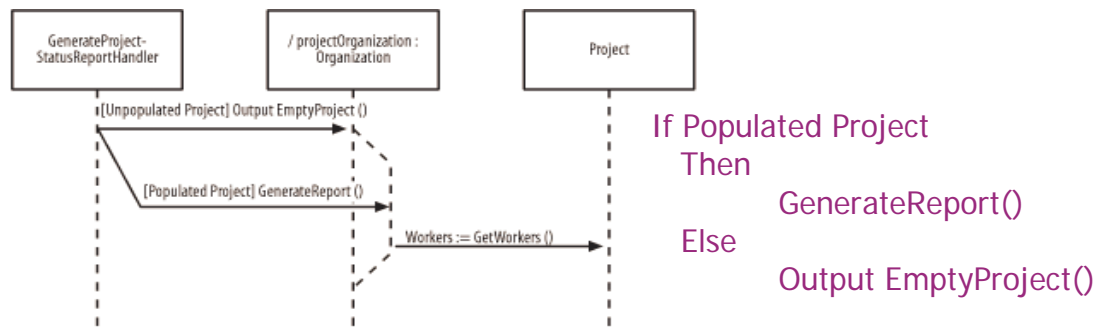
• Se pueden reproducir **estructuras de control**:

■ Iteraciones indefinidas:

- *[For each worker] (un "for" de C++)
- *[No more workers] (un "while" de C++)

■ Bifurcaciones:

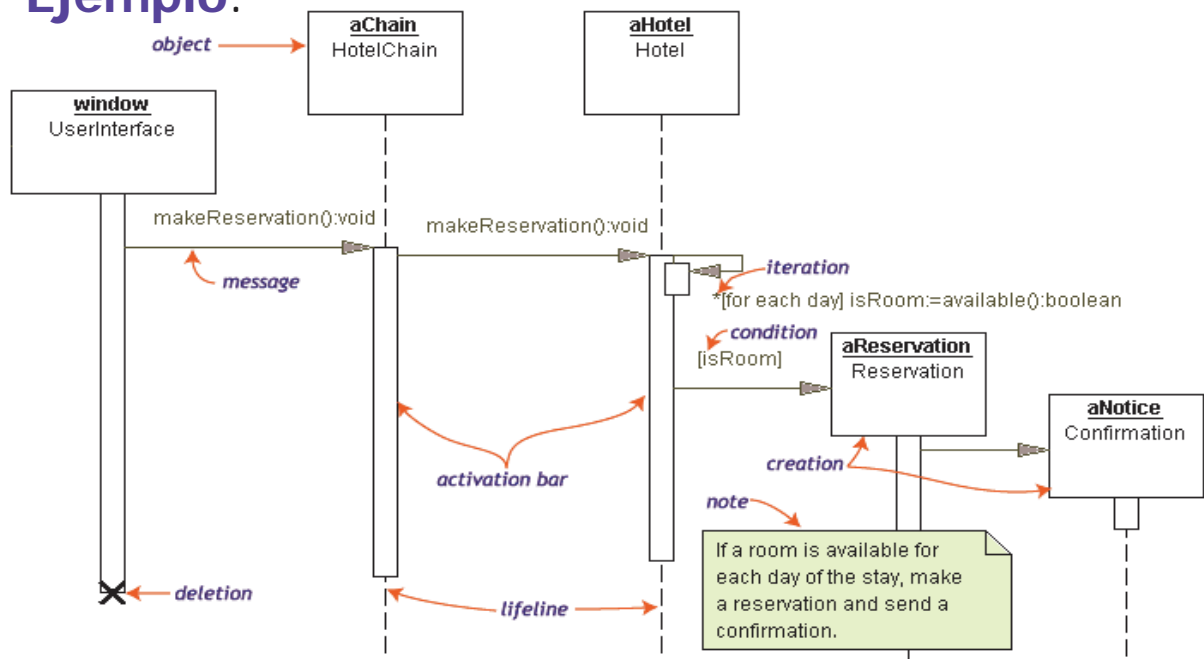
- Combinando condiciones de guarda con divisiones de líneas de vida.





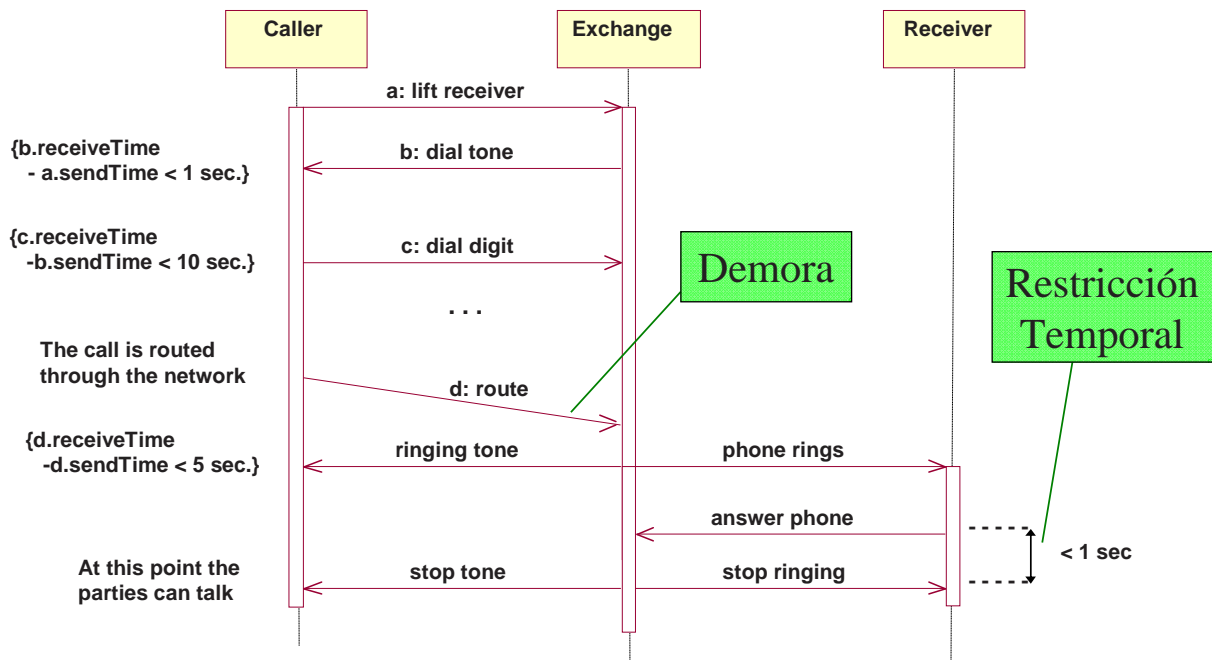
Diagramas de Interacción – De Secuencia

• Ejemplo:



Diagramas de Interacción – De Secuencia

• Ejemplo con demoras y restricciones temporales:





Diagramas de Interacción – De Secuencia

- **Ejemplo** de dos diagramas que se continúan:

Diagram 1

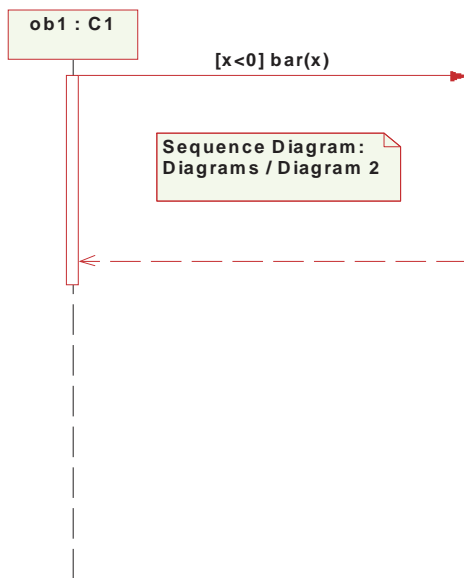
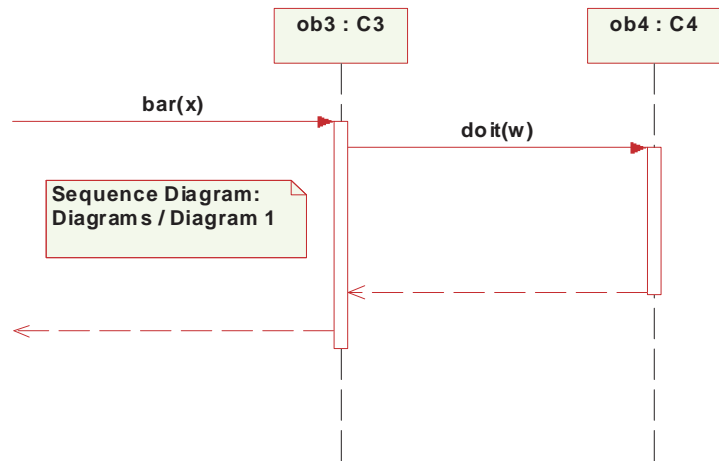


Diagram 2



Diagramas de Interacción – De Comunicación

- Un **Diagrama de Comunicación** es un diagrama de interacción que resalta la **organización estructural** de los objetos o roles que envían y reciben mensajes.
 - Muestran un conjunto de **roles**, enlaces entre ellos y los **mensajes** enviados y recibidos por las instancias que interpretan dichos roles.
 - La estructura estática viene dada por los enlaces; la dinámica por el envío de mensajes por los enlaces.



- En versiones anteriores a UML 2 se llamaban de colaboración.



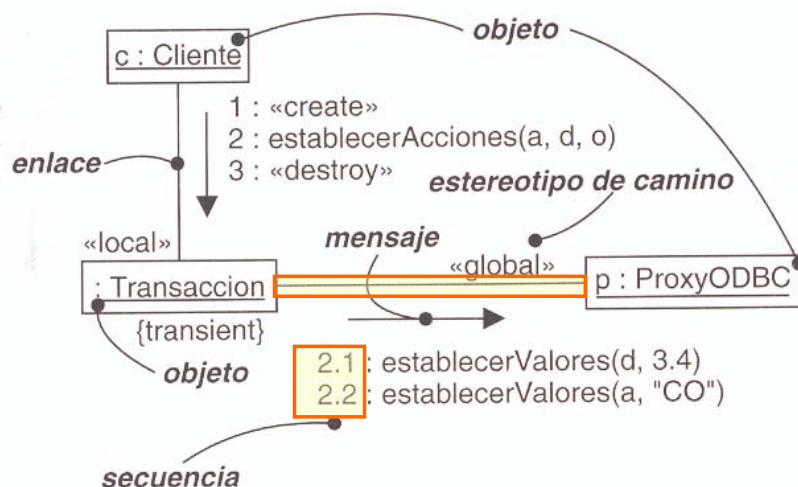
Diagramas de Interacción – De Comunicación

- Pasos para su elaboración:
 - 1) Colocar los **objetos** que participan en la interacción como nodos del grafo.
 - 2) Representar los **enlaces** que conectan esos objetos como arcos del grafo.
 - 3) Los enlaces se adornan con los **mensajes** que envían y reciben los objetos.
 - 4) **Anotar** las creaciones y destrucciones.
- Estos diagramas muestran de forma clara **el flujo de control** en el contexto de la organización estructural de los objetos que interaccionan.



Diagramas de Interacción – De Comunicación

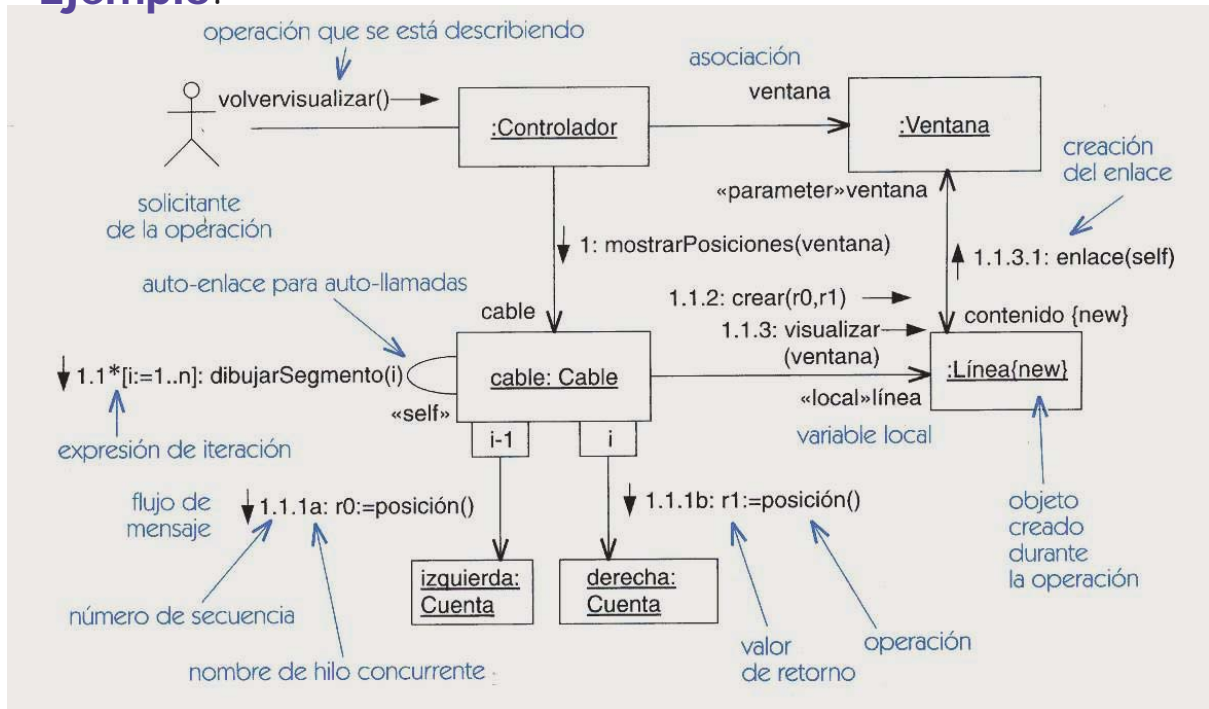
- Tienen dos características que los distinguen de los diagramas de secuencia:
 - El **Camino** para indicar cómo se enlaza un objeto a otro.
 - El **Número de Secuencia** para indicar la ordenación temporal de los mensajes.





Diagramas de Interacción – De Comunicación

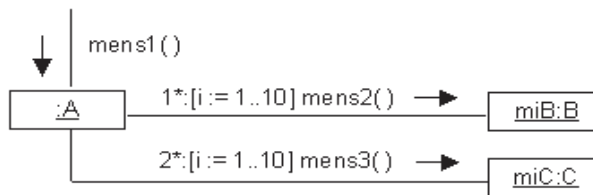
• Ejemplo.



Diagramas de Interacción – De Comunicación

• Se pueden reproducir **estructuras de control**:

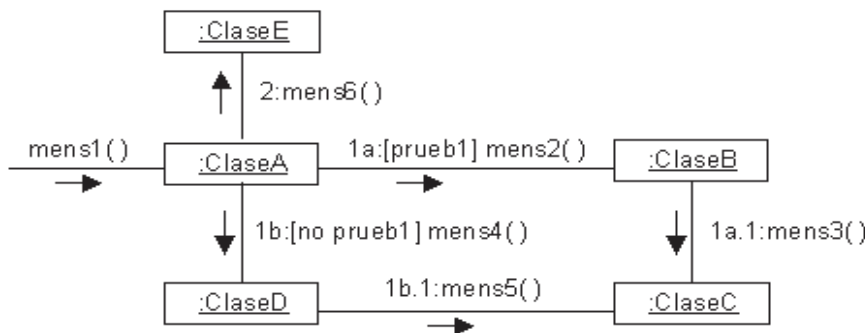
■ Iteraciones



```

msg1() {
  for i := 1 to 10 {
    miB.mens2();
    miC.mens3();
  }
}
  
```

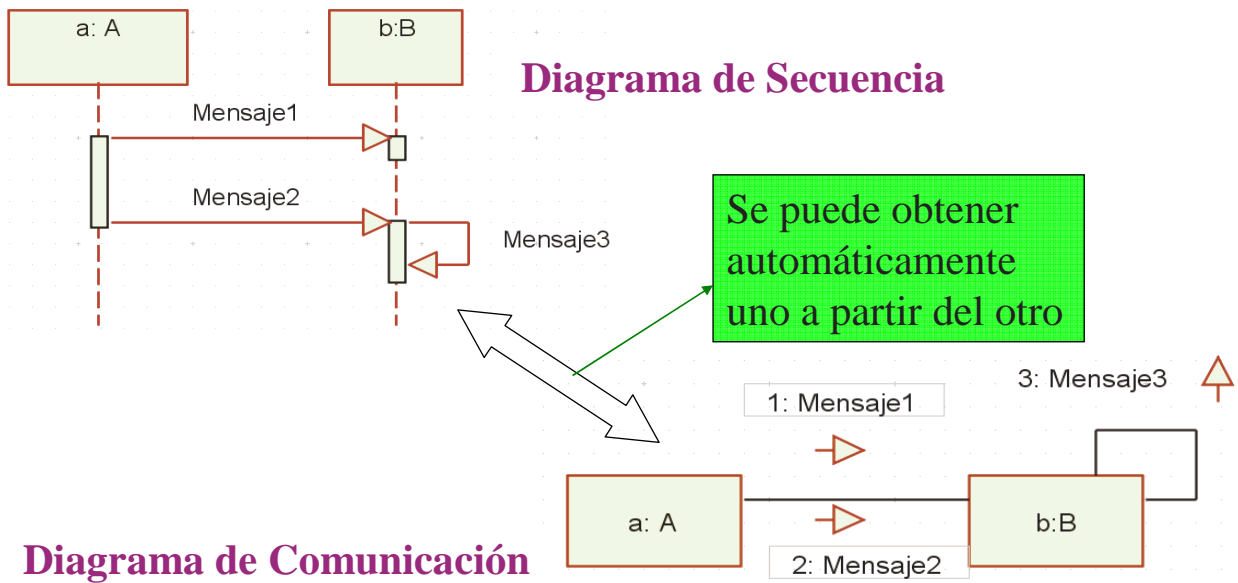
■ Bifurcaciones





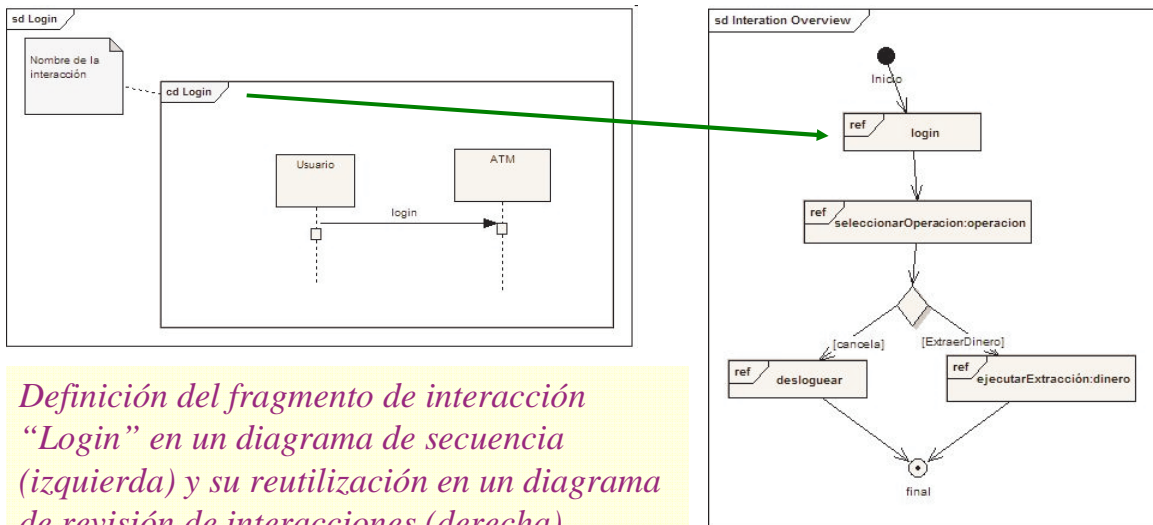
Diagramas de Interacción – De Comunicación

- Existe **equivalencia semántica** entre los diagramas de secuencia y los de comunicación.



Diagramas de Interacción – Control Estructurado

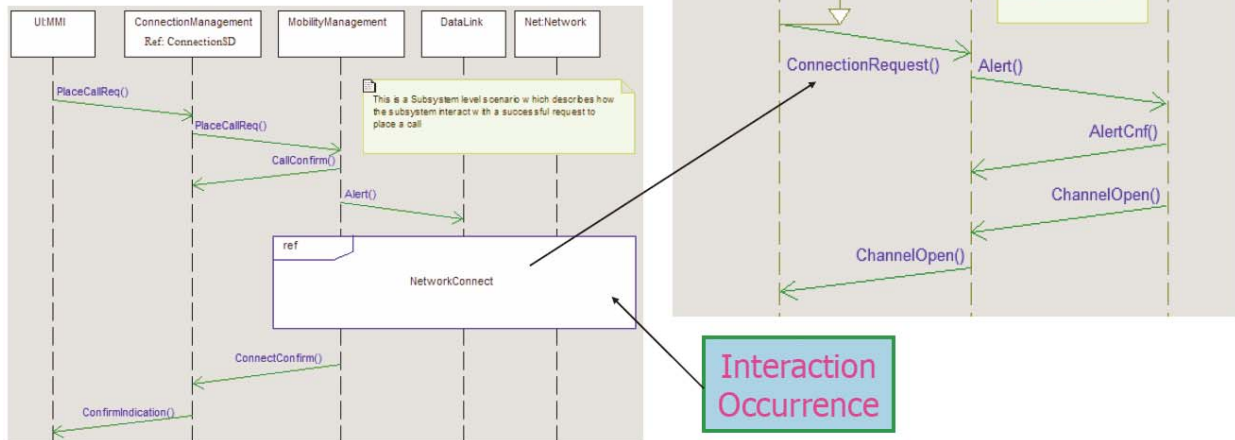
- En **UML 2** los 4 tipos de diagramas de interacción permiten **fragmentos de interacción** (interacciones reutilizables).
 - Sinónimo: fragmento combinado.





Diagramas de Interacción – Control Estructurado

El diagrama de secuencia de la izquierda incluye la referencia al fragmento de interacción “NetworkConnect”, que se define en el diagrama de la derecha.



Diagramas de Interacción – Control Estructurado

- Además, UML 2 ha añadido **operadores de interacción** a los diagramas de secuencia:
 - Se representan como regiones rectangulares dentro del diagrama.
 - Se indican en la etiqueta de la esquina superior izquierda.
 - El cuerpo del operador (alcance) son las líneas de vida que lo cruzan.



Diagramas de Interacción – Control Estructurado

- **Operadores de interacción**

- **Nombrar**

- **sd** – diagrama de secuencia con nombre
- **ref** – referencia a un fragmento de interacción

- **Flujo de Control**

- **loop** – repetir (iterar) un fragmento de interacción
- **opt** – opcional (se ejecuta si se cumple una condición)
- **alt** – selección entre varias alternativas (regiones separadas por líneas discontinuas horizontales). Existe el "else".
- **par** – concurrencia (paralelismo entre las regiones definidas)
- **break** – fragmento de ruptura elegido como alternativa al resto de la interacción



Diagramas de Interacción – Control Estructurado

- **Operadores de interacción (cont.)**

- **Orden**

- **seq** – ordenación débil (Secuenciación parcial)
- **strict** – ordenación estricta
- **critical** – fragmento atómico a efectos de validez

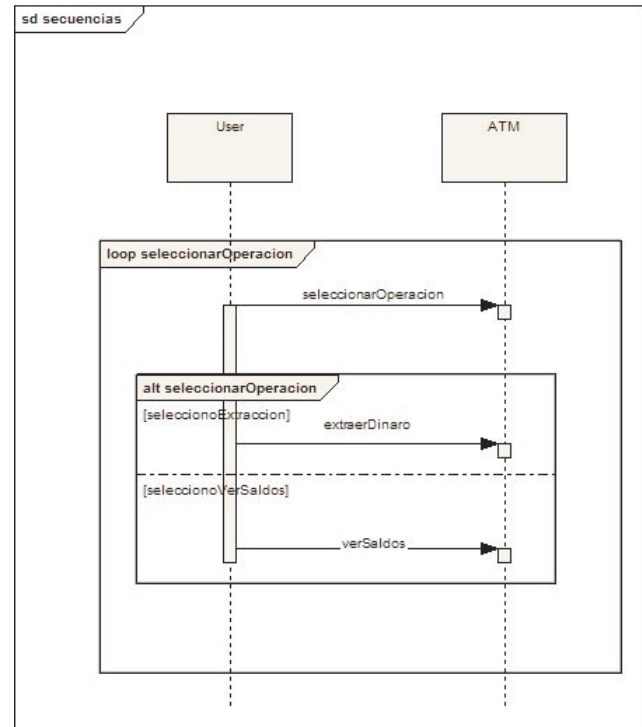
- **Causalidad**

- **assert** – requerido (el fragmento representa una aserción)
- **neg** – "no puede ocurrir" (especificación negativa)
- **ignore / consider** – listas de mensajes a ignorar o a considerar



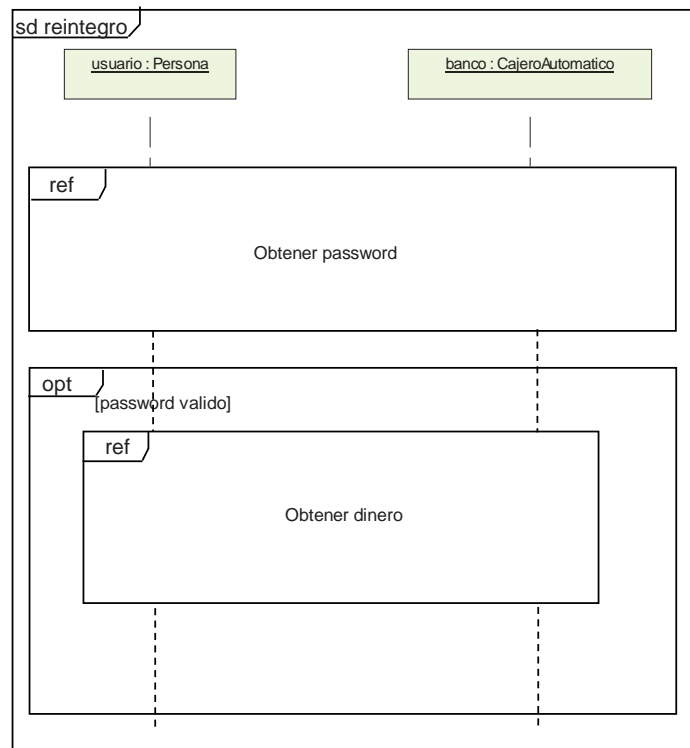
Diagramas de Interacción – Control Estructurado

- Operadores de Interacción
 - Ejemplo de **Loop y Alt**
 - *Iterar (loop) seleccionarOperación*
 - *En cada iteración elegir (alt) entre extraerDinero o verSaldo.*



Diagramas de Interacción – Control Estructurado

- Los diagramas de secuencia demasiado grandes pueden simplificarse dividiéndolos mediante el empleo del operador **ref**.
- Se pueden crear tantos **niveles de anidamiento** como interese.





Diagramas de Interacción - Consejos

- Una **interacción bien estructurada**:
 - **Sencilla** =>
 - Incluye sólo aquellos objetos que colaboran para llevar a cabo un comportamiento mayor que la suma de los comportamientos de sus elementos.
 - **Contexto claro** =>
 - Un sistema, clase u operación.
 - **Eficiente** =>
 - Lleva a cabo su comportamiento con equilibrio de tiempo y recursos.
 - **Adaptable** =>
 - Los elementos mas proclives a cambiar se deben aislar para modificarlos más fácilmente.
 - **Comprensible** =>
 - No incluir trucos o efectos laterales ocultos. Tener una semántica sencilla.



Diagramas de Interacción - Consejos

- Un **diagrama de interacción bien estructurado**:
 - Se ocupa de modelar un aspecto de la dinámica de un sistema.
 - Contiene sólo aquellos elementos esenciales para comprender ese aspecto.
 - Proporciona detalles de forma consistente con su nivel de abstracción y sólo debe mostrar aquellos adornos que son esenciales para su comprensión.
 - Debe proveer información suficiente para entender los aspectos importantes de la semántica de la interacción.



Diagramas de Interacción - Consejos

- Al **dibujar** una **interacción** en UML:
 - Elegir el aspecto a destacar en la interacción:
 - **Ordenación temporal** de los mensajes
 - -> Diagrama de **Secuencia**
 - **Organización de los objetos** implicados
 - -> Diagrama de **Comunicación**
 - Recordar que los eventos en sub-secuencias separadas sólo están parcialmente ordenados.
 - Cada sub-secuencia está ordenada, pero los tiempos relativos entre eventos de sub-secuencias diferentes no están fijados.
 - Mostrar solo las propiedades importantes para comprender la interacción en su contexto:
 - Objetos: valores de atributos, rol y estado.
 - Mensajes: parámetros, semántica de ocurrencia, valor de retorno.



Diagramas de Interacción - Consejos

- Al **dibujar** un **diagrama de interacción** en UML:
 - Asignarle un **nombre** que comunique su propósito.
 - Elegir un **tipo de diagrama**:
 - De Secuencia (primar ordenación temporal de mensajes).
 - De Comunicación (primar organización de los objetos).
 - Distribuir sus elementos para **minimizar los cruces** de líneas.
 - Utilizar **notas y colores** como señales visuales para llamar la atención sobre las características importantes del diagrama.
 - Usar la **bifurcación** de forma moderada.
 - Las bifurcaciones complejas se pueden representar mucho mejor con los diagramas de actividad.



Modelado

- Las **interacciones** suelen utilizarse con el propósito de **modelar el flujo de control** que caracteriza el **comportamiento** de
 - un sistema (casos de uso, patrones, mecanismos y frameworks) , o
 - una clase u operación individual.
- Cuando se modela una interacción se construye una historia de las acciones que tienen lugar entre un conjunto de objetos.
 - Las **tarjetas CRC** son útiles para ayudar a descubrir y razonar tales interacciones.



Modelado

- Cuando se modelan los **aspectos dinámicos de un sistema, subsistema, operación o clase**, se pueden utilizar los diagramas de interacción siguiendo dos estrategias diferentes:
 - Modelado de **Flujos de Control por Ordenación Temporal** con **diagramas de secuencia**.
 - Modelado de **Flujos de Control por Organización** con **diagramas de comunicación**.
- También se pueden usar diagramas de interacción para:
 - Modelar un **escenario de un caso de uso**.
 - Modelar los aspectos dinámicos de una sociedad de objetos (**colaboración**).



Modelado – Flujo de Control por Tiempos

- Para modelar un **flujo de control por ordenación temporal** (diagrama de secuencia):
 1. Establecer el **contexto de la interacción**.
 - Sistema, subsistema, operación o una clase; o
 - Escenario de un caso de uso o de una colaboración.
 2. Establecer el **escenario de la interacción**.
 - Identificando los objetos que juegan un rol en ella.
 - Mas importantes a la izquierda.
 3. Establecer la **línea de vida** de cada objeto.
 - Indicar creaciones y destrucciones de objetos.
 4. A partir del mensaje que inicia la interacción:
 - **Colocar los mensajes** subsiguientes desde arriba hacia abajo.
 - Mostrar las propiedades de cada mensaje según sea necesario para explicar la semántica de la interacción.



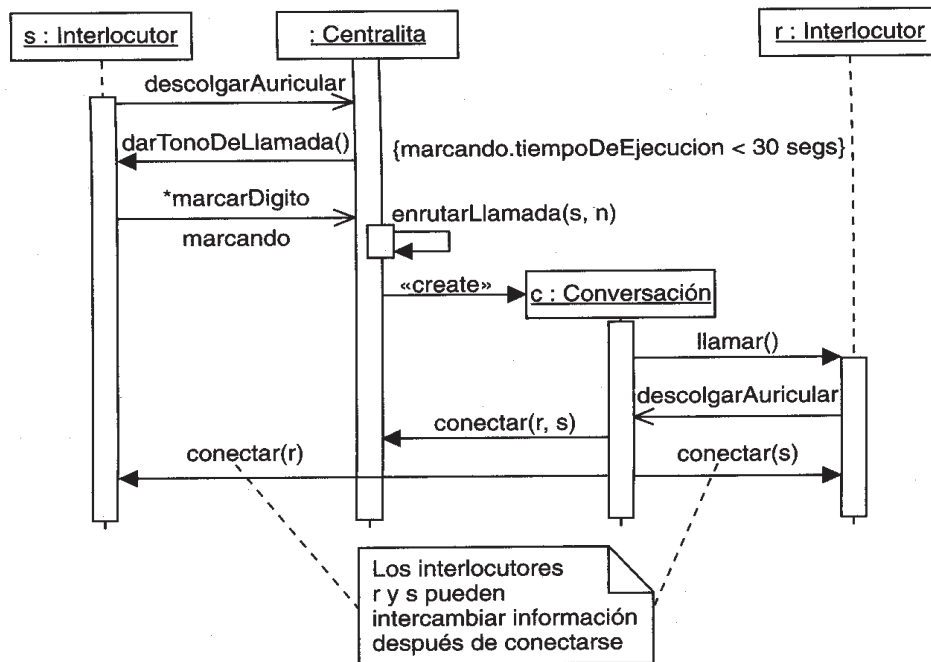
Modelado – Flujo de Control por Tiempos

- Para modelar un **flujo de control por ordenación temporal** (diagrama de secuencia) (cont):
 5. Si es necesario:
 - Visualizar el anidamiento de mensajes o el intervalo de tiempo en el que tiene lugar la computación:
 - Adornar la línea de vida de cada objeto con su **foco de control**.
 - Especificar restricciones de tiempo o espacio:
 - Adornar cada mensaje con una **marca de tiempo** y asociar las **restricciones** apropiadas.
 - Especificar más formalmente el flujo de control:
 - Asociar **pre y poscondiciones** a cada mensaje.



Modelado – Flujo de Control por Tiempos

• Ejemplo.



Modelado – Flujo de Control por Organización

• Para modelar un **flujo de control por organización** (diagrama de comunicación):

1. Establecer el **contexto de la interacción**.
 - Sistema, subsistema, operación o una clase; o
 - Escenario de un caso de uso o de una colaboración.
2. Establecer el **escenario de la interacción**.
 - Identificando los objetos que juegan un rol en ella.
 - Poner importantes en el centro, secundarios en el exterior.
3. Especificar los **enlaces** entre objetos.
 - Primero los de asociaciones.
 - Adornarlos con las anotaciones (global, etc.) adecuadas.



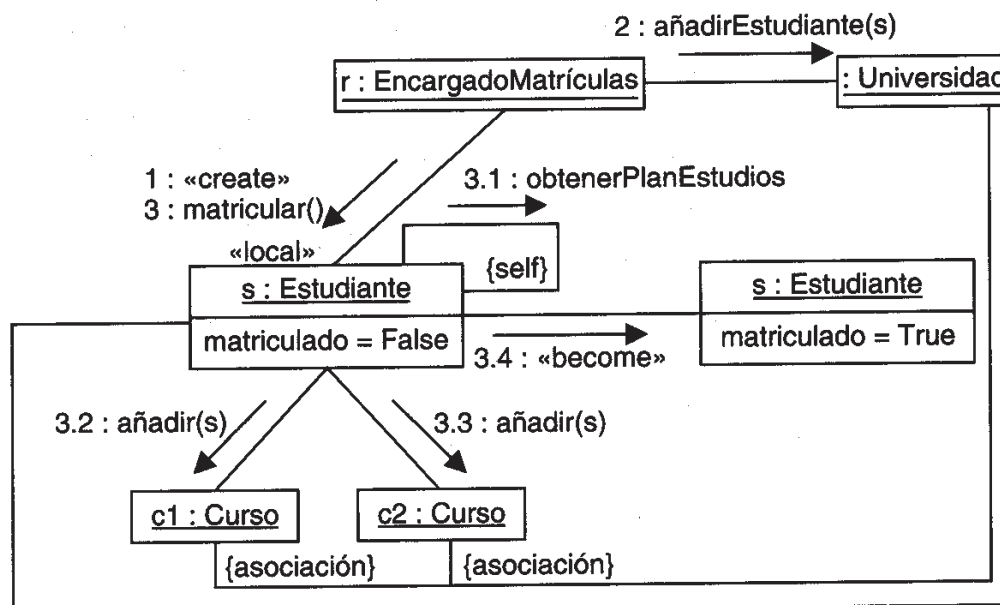
Modelado – Flujo de Control por Organización

- Para modelar un **flujo de control por organización** (diagrama de comunicación) (cont):
 4. A partir del mensaje que inicia la interacción:
 - **Asociar cada mensaje al enlace** apropiado, estableciendo su número de secuencia.
 - Representar los anidamientos con numeración decimal.
 5. Si es necesario:
 - Especificar restricciones de tiempo o espacio:
 - Adornar cada mensaje con una **marca de tiempo** y asociar las **restricciones** apropiadas.
 - Especificar más formalmente el flujo de control:
 - Asociar **pre y poscondiciones** a cada mensaje.



Modelado – Flujo de Control por Organización

• Ejemplo.





Modelado – Requisitos vs Diseño

- Los diagramas de interacción se utilizan de dos maneras diferentes, según sea la **fase** del ciclo de vida y el **nivel de detalle** deseado:
 - **REQUISITOS**: Documentar los escenarios de un caso de uso.
 - Los **casos de uso** indican cómo los **actores** interactúan con el **sistema**.
 - **DISEÑO**: Representar de forma precisa las interacciones entre los objetos.
 - Durante la **operación del sistema**, los actores generan **eventos**, solicitando alguna operación a cambio.



Modelado – Requisitos vs Diseño

- Durante la etapa de Requisitos
 - El diagrama de interacción **documenta un determinado escenario de un caso de uso**, indicando los **eventos generados por actores externos**, su orden y los **eventos de respuesta** del sistema.
 - Es una descripción de la interacción, en términos próximos al usuario y sin entrar en detalles de sincronización.
 - Las flechas no corresponden aún a envíos de mensajes en el sentido de lenguajes de programación.
 - No se establece distinción entre flujos de control y de datos.



Modelado – Requisitos vs Diseño

- Es posible la generación automática del diagrama de secuencia desde la especificación de un caso de uso.

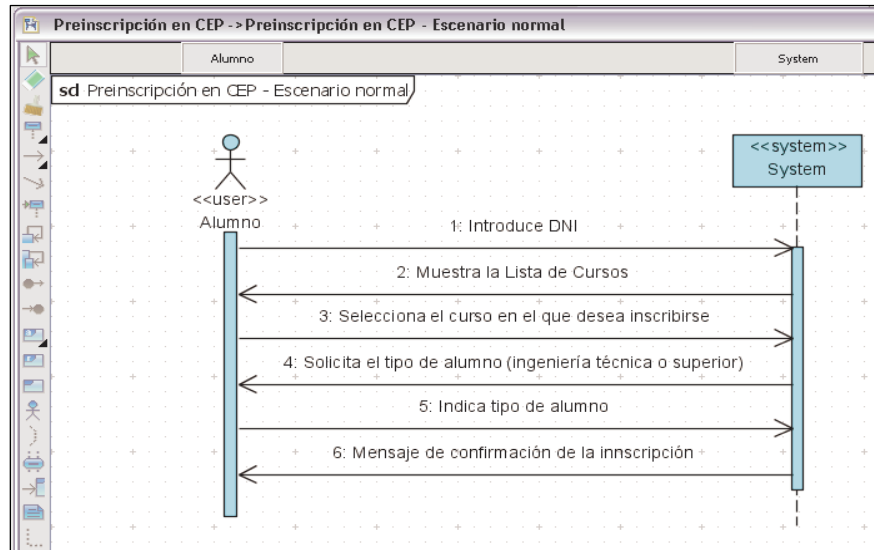


Diagrama de secuencia generado con Visual Paradigm a partir de la especificación del caso de uso (use case detail).



Modelado – Requisitos vs Diseño

- Durante el Diseño
 - El diagrama de interacción es una **representación precisa de las interacciones entre los objetos**.
 - Realización de casos de uso.
 - El concepto de mensaje permite unificar todas las formas de comunicación entre objetos.
 - Llamadas a procedimientos, eventos discretos, señales entre flujos de ejecución, o interrupciones de hardware.



Modelado – Requisitos vs Diseño

- **Ejemplo**

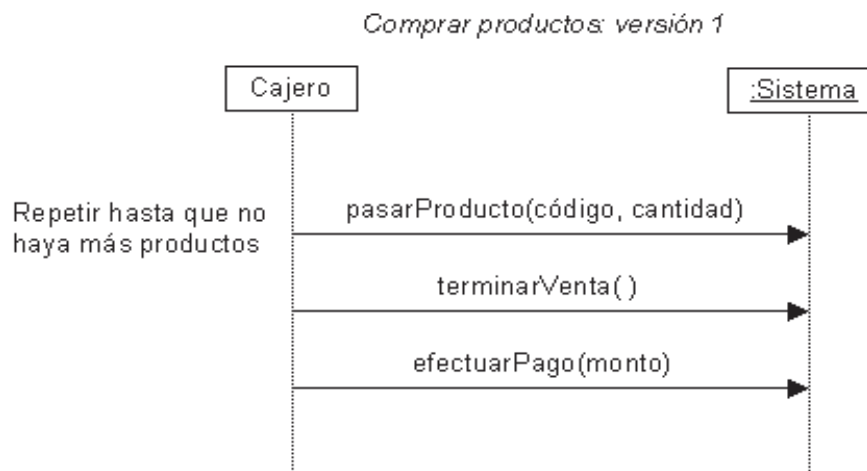
- Caso de uso para compra de productos en un supermercado.
- **Caso de uso:** Comprar productos
- **Actores:** Cliente, Cajero
- **Tipo:** Primario
- **Descripción:** Un Cliente llega a la caja registradora con los artículos que va a comprar. El Cajero registra los artículos y cobra el importe. Al terminar la operación, el Cliente se marcha con los productos y con el recibo de la compra.



Modelado – Requisitos vs Diseño

- **Ejemplo (cont.)**

- Diagrama de secuencia para modelar el escenario principal del caso de uso (**requisitos**).





Modelado – Requisitos vs Diseño

- Ejemplo (cont.)
 - Diagrama de comunicación para detallar el evento “efectuarPago” (**diseño**):

