



INGENIERÍA DEL SOFTWARE I

Tema 7

Interacciones del Sistema
(en desarrollo OO)

Univ. Cantabria – Fac. de Ciencias
Patricia López y Francisco Ruiz



Objetivos del Tema

- Objetivos
 - Conocer en detalle los conceptos de **Interacciones**.
 - Aprender a modelar **flujos de control** en UML 2
 - Por ordenación temporal de mensajes => **Diagramas de secuencia**
 - Por organización de objetos => **Diagramas de comunicación**
 - Comprender los **diferentes usos** de los diagramas de interacción en las fases de **requisitos** y **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.
 - Miles y Hamilton (2006): Learning UML 2.0
 - Caps. 7 y 8,



Contenido

- Introducción
- Conceptos de Interacciones
 - Interacción
 - Objeto y Rol
 - Enlace
 - Mensaje
 - Secuenciación
- Diagramas de Interacción
 - De Secuencia
 - Control Estructurado
 - De Comunicación
 - Equivalencia
- Modelado
 - Requisitos vs Diseño
- Ejercicios



Introducción

- Las interacciones se utilizan para modelar los **aspectos dinámicos de un sistema**.
- En cualquier **sistema OO** los **objetos** interactúan entre sí intercambiando **mensajes**.
- Una **interacción** es un comportamiento
 - que incluye un conjunto de **mensajes**
 - que intercambian entre sí un conjunto de **objetos**
 - dentro de un **contexto**
 - para lograr un **propósito**.
- Las interacciones son un mecanismo común para la descripción de un sistema a todos los niveles
 - Lo pueden utilizar tanto los desarrolladores como los usuarios



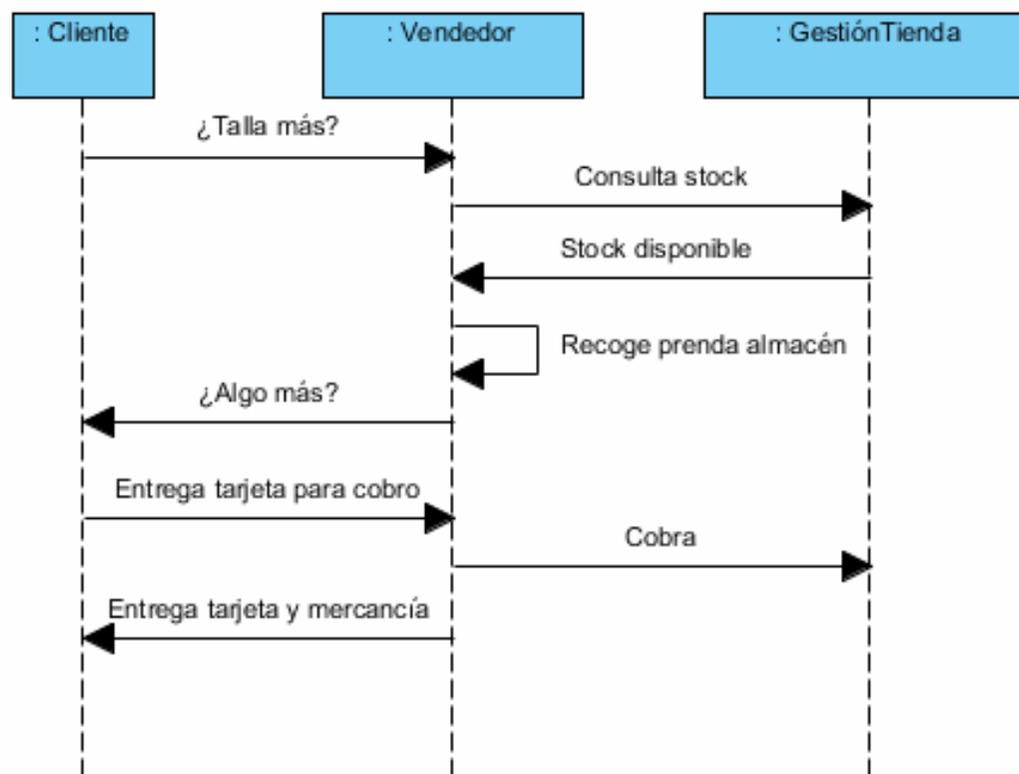
Conceptos de Interacciones

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



Conceptos - Interacción

- Una interacción modela un **escenario** concreto, presentando:
 - Todos los objetos que colaboran
 - Los mensajes enviados entre los objetos.
- Las interacciones se centran en los **mensajes** intercambiados entre los objetos, y no en los datos asociados a esos mensajes.





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.
 - 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**.
 - Cualquier **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.

pepe:Cliente

juan:Cliente

- Elementos **prototípicos**: (ROLES)

- Representan cualquier elemento de un cierto tipo.

:Cliente

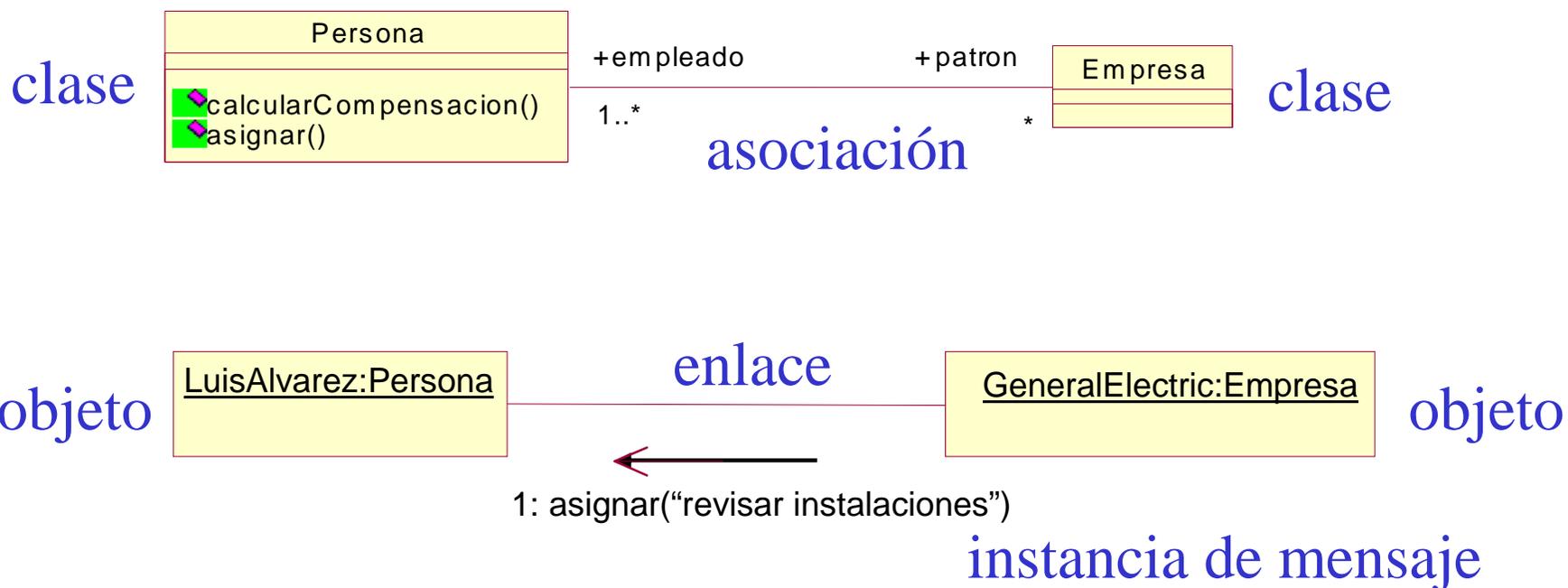
- En el contexto de una interacción se pueden encontrar **instancias de**:
 - Clases, componentes, nodos y casos de uso.
 - Las instancias de **clases abstractas** y de **interfaces** en realidad representan a instancias indirectas (prototípicas o roles) de cualquier clase, hija de la clase abstracta o que realiza la interfaz.



Conceptos – Enlace

- **Enlaces:**

- Conexiones entre objetos => Instancias de una asociación.
- Especifican un camino a lo largo del cual un objeto puede enviar un mensaje a otro objeto (o a sí mismo).

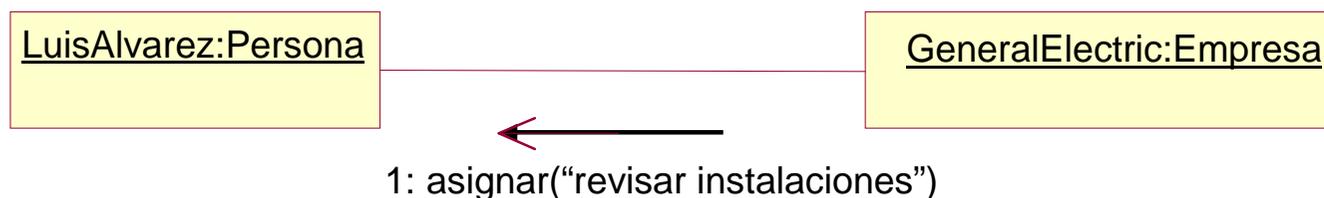




Conceptos – Mensaje

- **Mensajes:**

- Especificación de una **comunicación entre objetos** a través de la que se **transmite una información** con la expectativa de que se desencadenará una actividad.



- El **envío** y la **recepción** de una instancia de un mensaje es una ocurrencia de un evento (*event occurrence* en UML).
- 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 Mensajes:**

- **Llamada:** Invoca una **operación**.

- Mensaje **síncrono**: el emisor espera hasta recibir el resultado.



- **Envío:** Envía una **señal**.

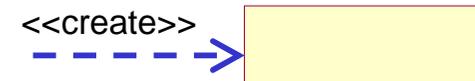
- Mensaje **asíncrono**: el emisor no espera a recibir el resultado.



- **Retorno:** Devuelve un valor al emisor.



- **Creación:** Crea un objeto.



- **Destrucción:** Destruye un objeto. Un objeto puede destruirse a sí mismo.



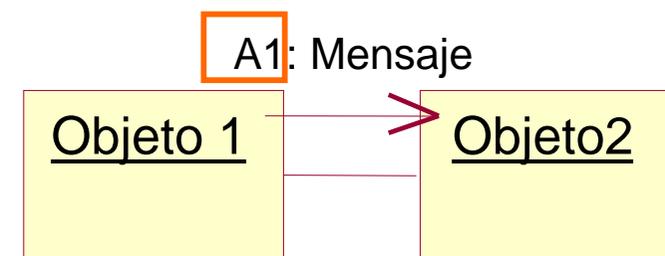
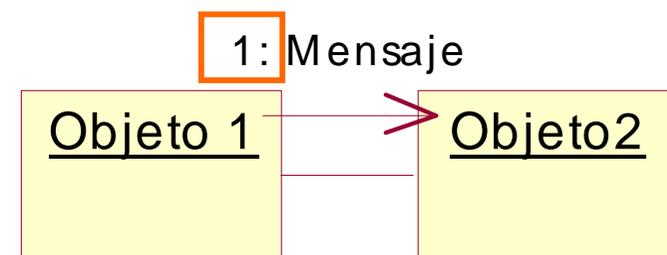
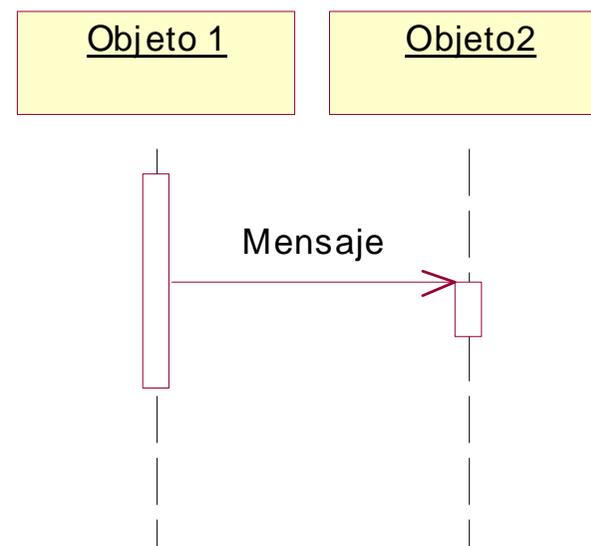
- Las llamadas y envíos pueden incluir **argumentos**

- Los retornos pueden incluir el **valor de retorno**.



Conceptos - Secuenciación

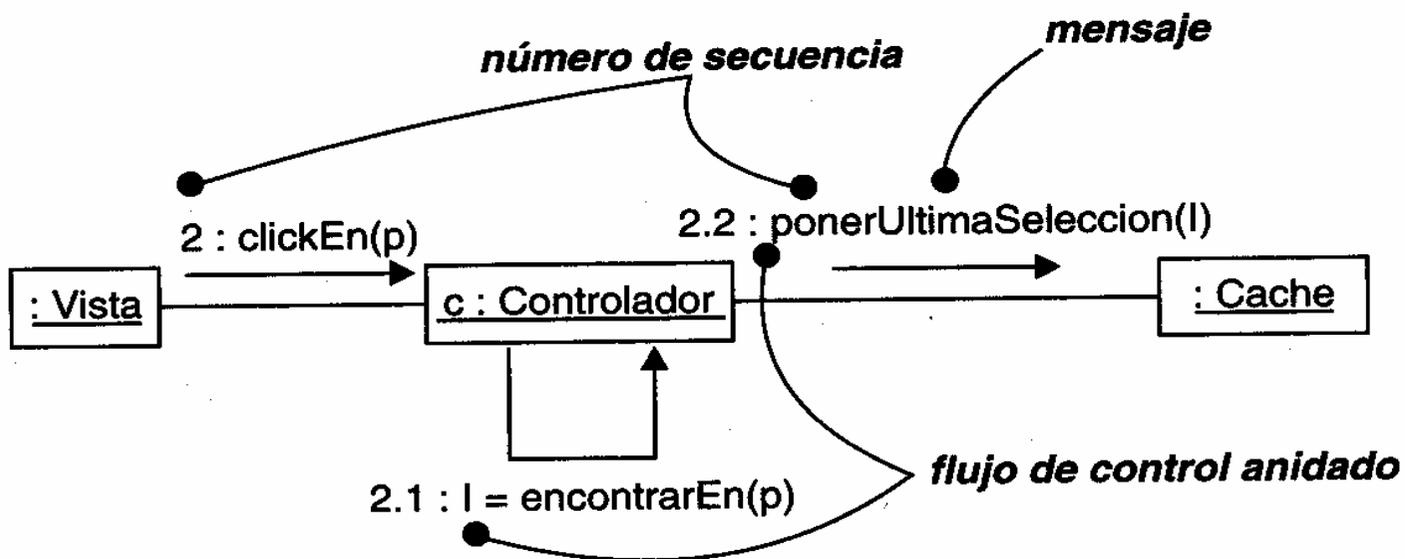
- Una **secuenciación** es una **secuencia de mensajes entre objetos**.
 - Cada proceso e hilo de un sistema define un flujo de control separado, y dentro de cada flujo los mensajes se ordenan en secuencia temporal.
 - Para mejorar la visualización:
 - Expresar la posición de un mensaje con relación al inicio de la secuencia
 - $\langle n^{\circ} \text{secuencia} \rangle: \langle \text{mensaje} \rangle$
 - Cuando hay varios flujos de control:
 - Identificar el proceso o hilo que envía cada mensaje
 - $\langle \text{id flujo} \rangle \langle n^{\circ} \text{secuencia} \rangle: \langle \text{mensaje} \rangle$





Conceptos - Secuenciación

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



- Plano** (sin anidamiento)





Diagramas de Interacción

- **RECORDATORIO (Tema 2)**
 - 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.
 - UML 2 incluye los siguientes
 - **Secuencia**
 - **Comunicación** (antiguo de Colaboración en UML 1.x)
 - **Tiempos**
 - **Revisión de las Interacciones**
 - **Contenido**:
 - Objetos o roles (Instancias de clases, interfaces, componentes y nodos)
 - Actores
 - Enlaces
 - Mensajes
 - Opcionalmente pueden incluir anotaciones (comentarios y restricciones)



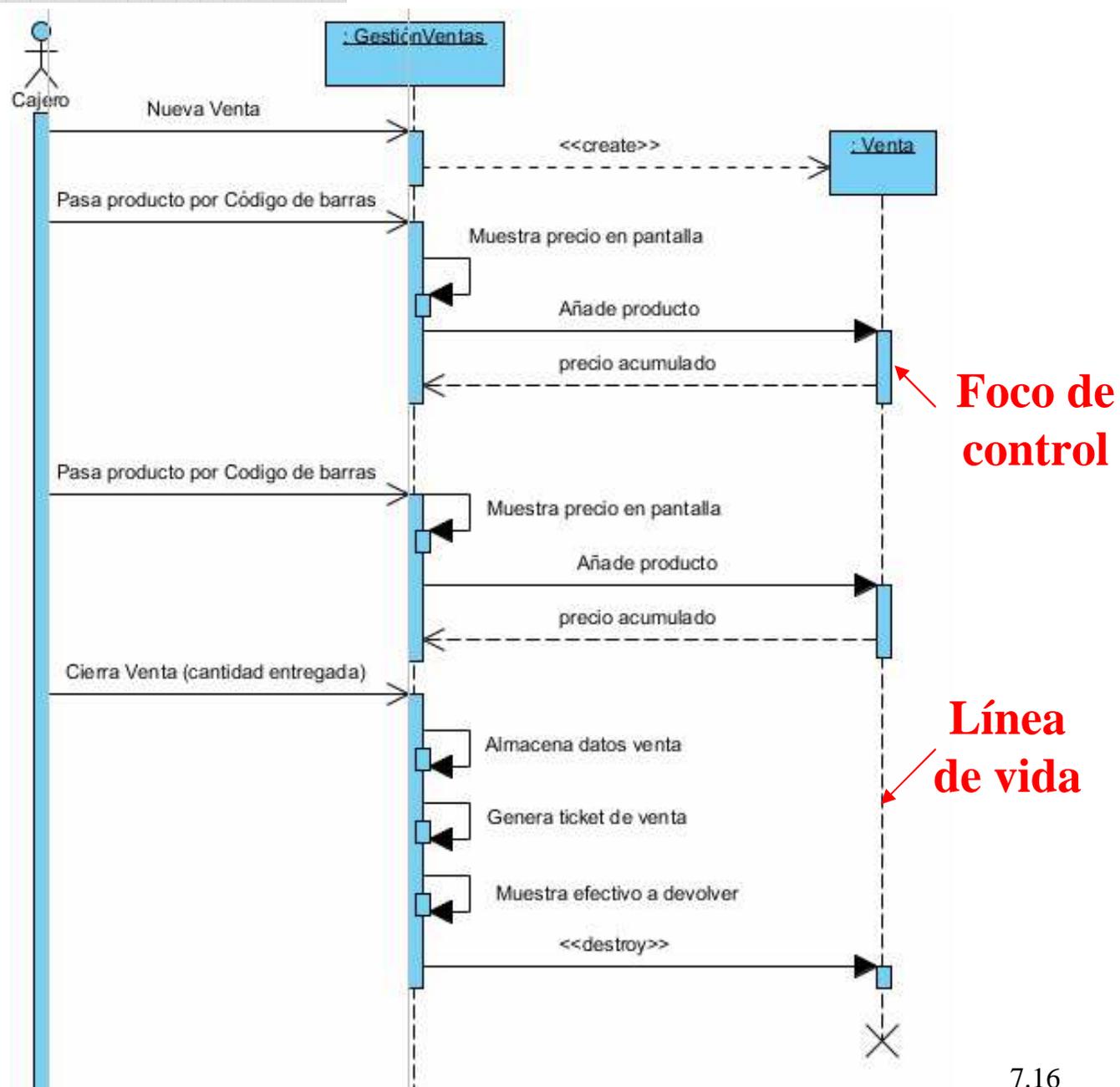
Diagramas de Interacción – Diagramas de Secuencia

- 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.
 - Con la línea de vida se puede indicar la **creación o destrucción de objetos** o roles durante la interacción:
 - El **Foco de Control** o **Barra de activación**
 - 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.
 - Son opcionales
 - 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 – Diagramas de Secuencia

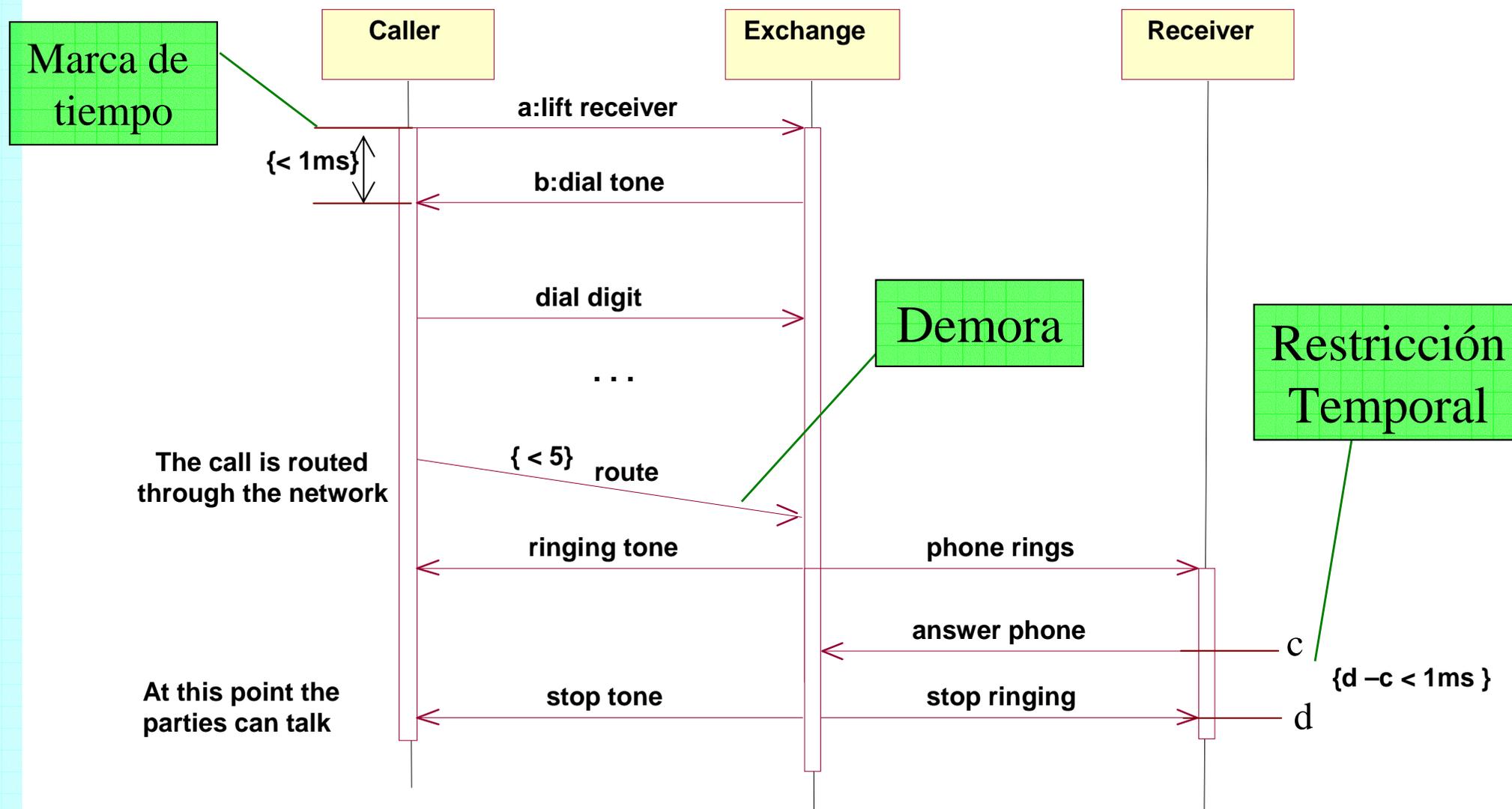
- Diagrama de secuencia que modela la compra de dos productos en el supermercado con pago en efectivo





Diagramas de Interacción – Diagramas de Secuencia

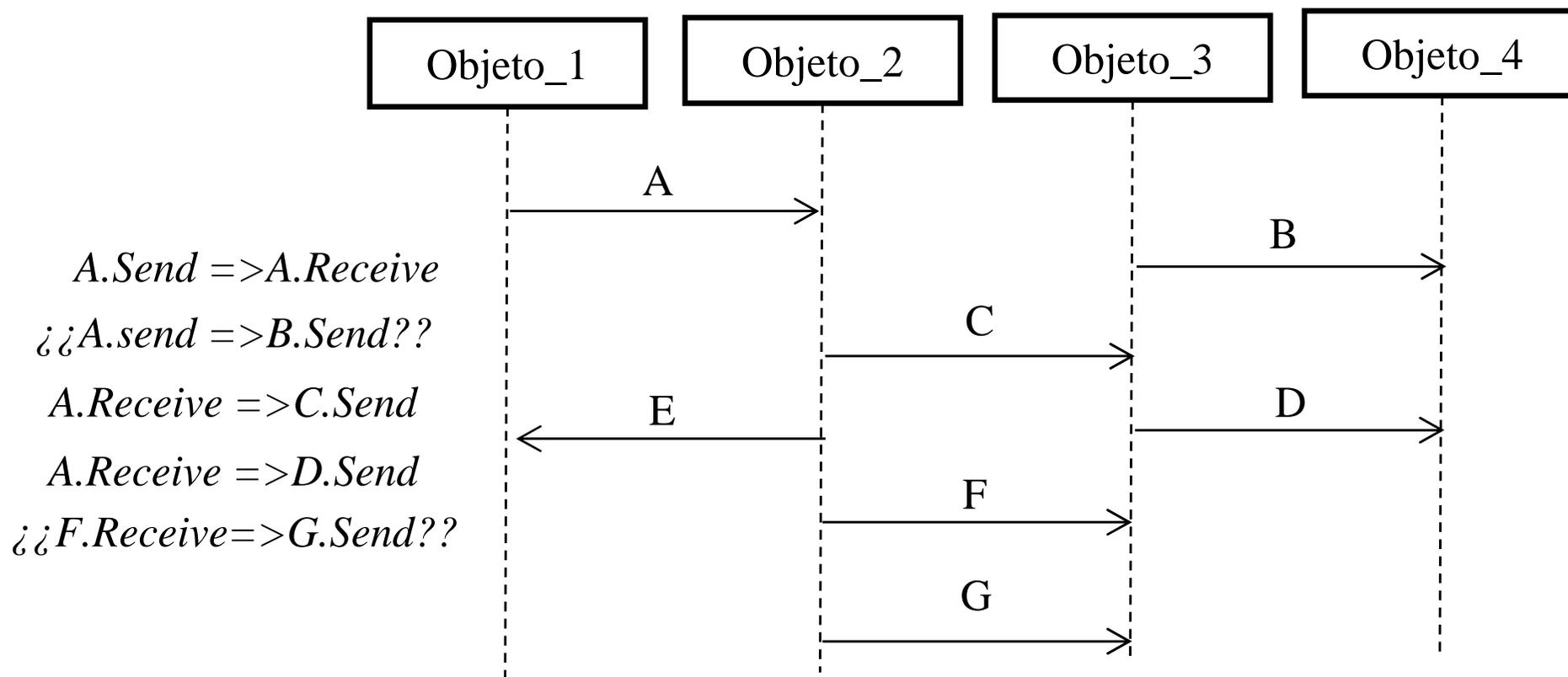
- **Ejemplo** con demoras y restricciones temporales:





Diagramas de Interacción – Diagramas de Secuencia

- Cuidado con las referencias temporales





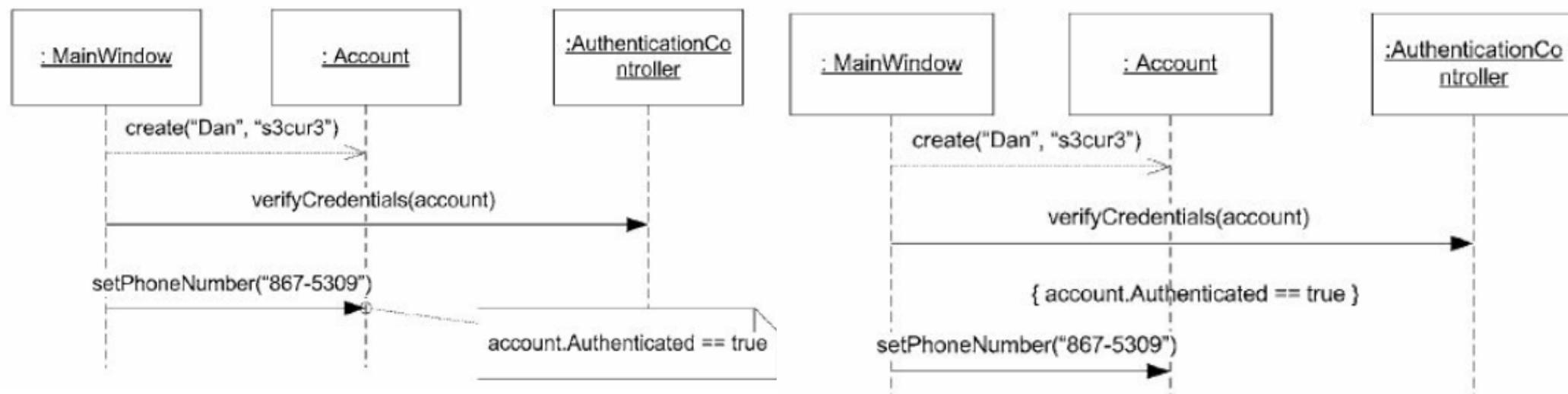
Diagramas de interacción – Diagramas de Secuencia

- **Sintaxis** para mensajes en diagramas de secuencia:
[numeroSecuencia :] [<atributo> =]
<nombre_mensaje>[(<argumentos>)] [: valorRetorno]
- **<atributo>**: representa el atributo de la clase invocante (o atributos globales definidos para la propia interacción) en la que se almacena el valor de retorno de la invocación
- Ejemplos:
 - *mymessage(14, -, 3.14, "hello")*
 - *v=mymsg(16, variab):96*
 - *mymsg(myint=16)*
- Solo el nombre del mensaje es obligatorio.
- En diagramas de secuencia no se suele utilizar el número de secuencia



Diagramas de interacción – Diagramas de Secuencia

- **Invariantes de estado** en diagramas de secuencia
 - Condiciones que tienen que cumplirse en un determinado punto del diagrama



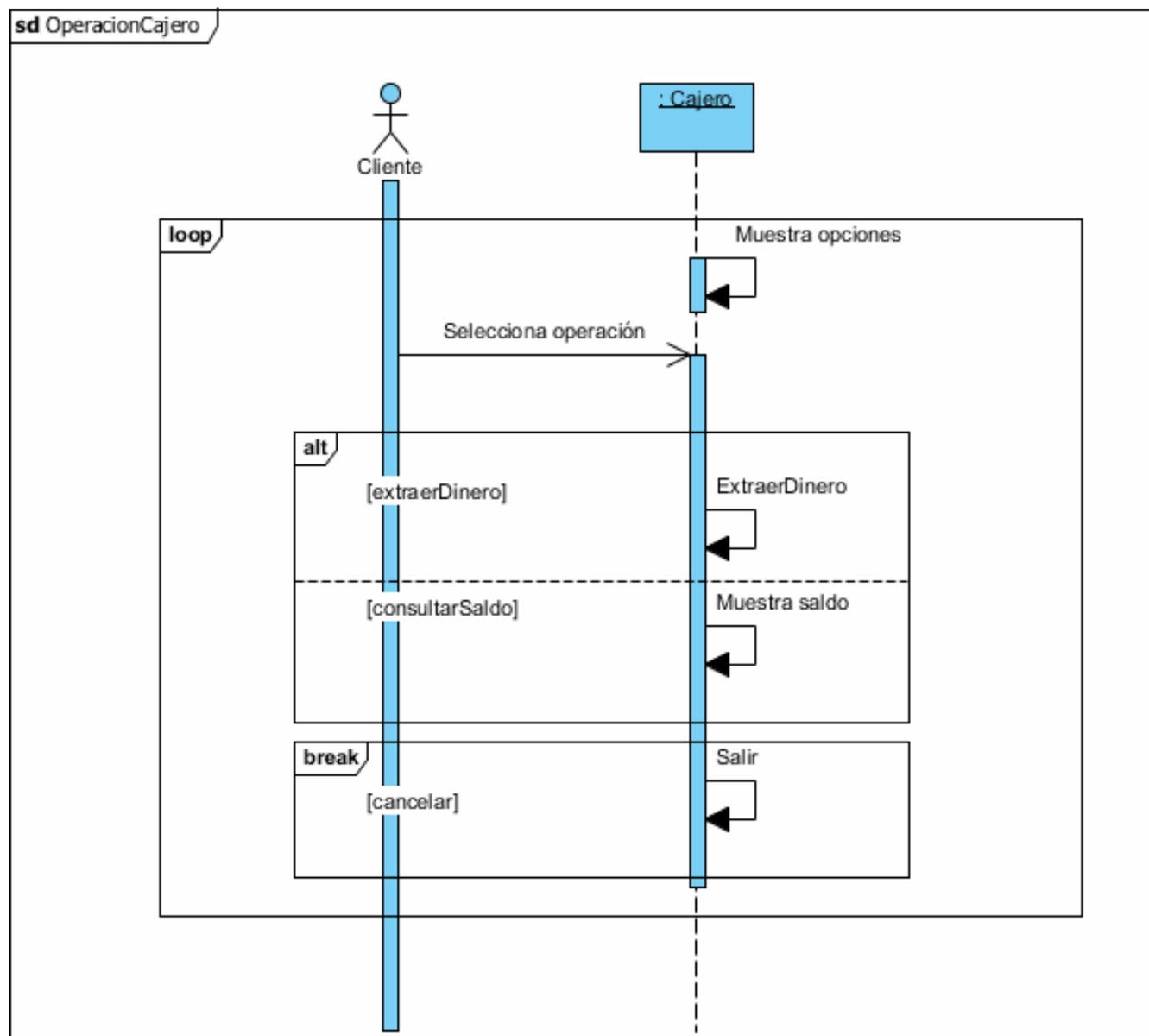


Diagramas de Interacción – Control Estructurado

- En **UML 2**, los flujos de control complejos en diagramas de secuencia se describen utilizando **fragmentos de interacción**:
 - Regiones rectangulares dentro del diagrama que engloban parte de las interacciones del diagrama.
 - El tipo de operador se indica en la etiqueta de la esquina superior izquierda.
 - El alcance del operador lo representan las líneas de vida que abarca.
 - Se denominan también **fragmentos combinados**.
 - **Operadores de interacción principales**:
 - **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).
 - **par** – concurrencia (paralelismo entre las regiones definidas)
 - **break** – fragmento de ruptura elegido como alternativa al resto de la interacción
 - **Orden**: seq, strict, critical
 - **Causalidad**: assert, neg, ignore



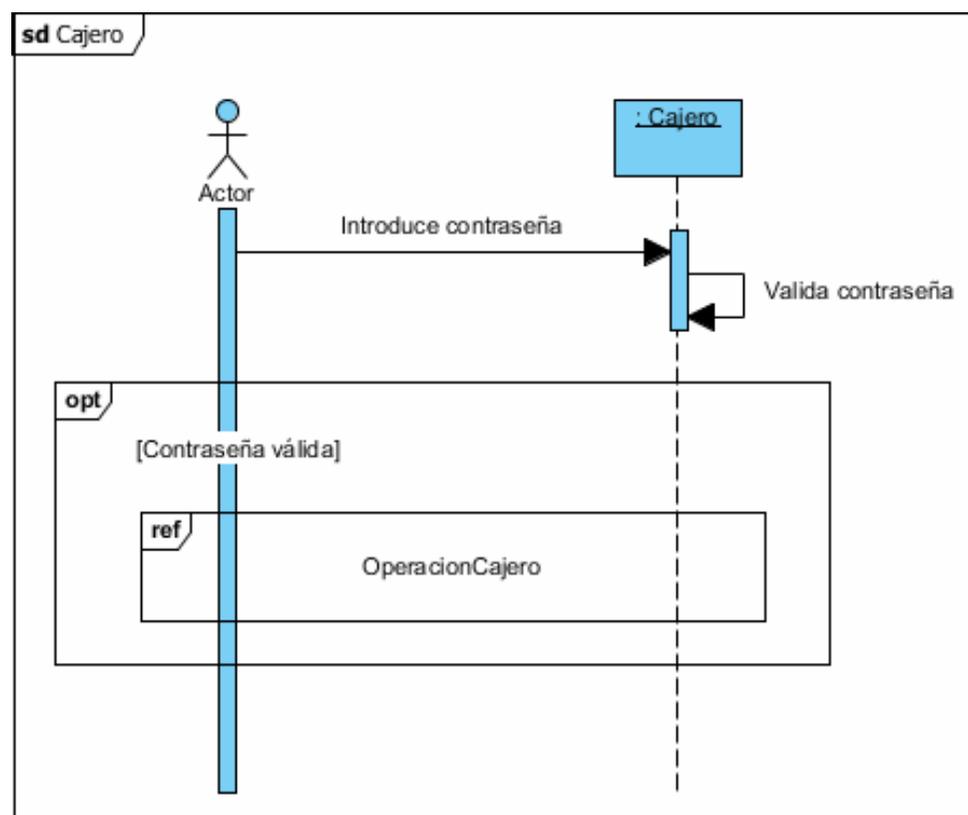
Diagramas de Interacción– Control Estructurado





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.

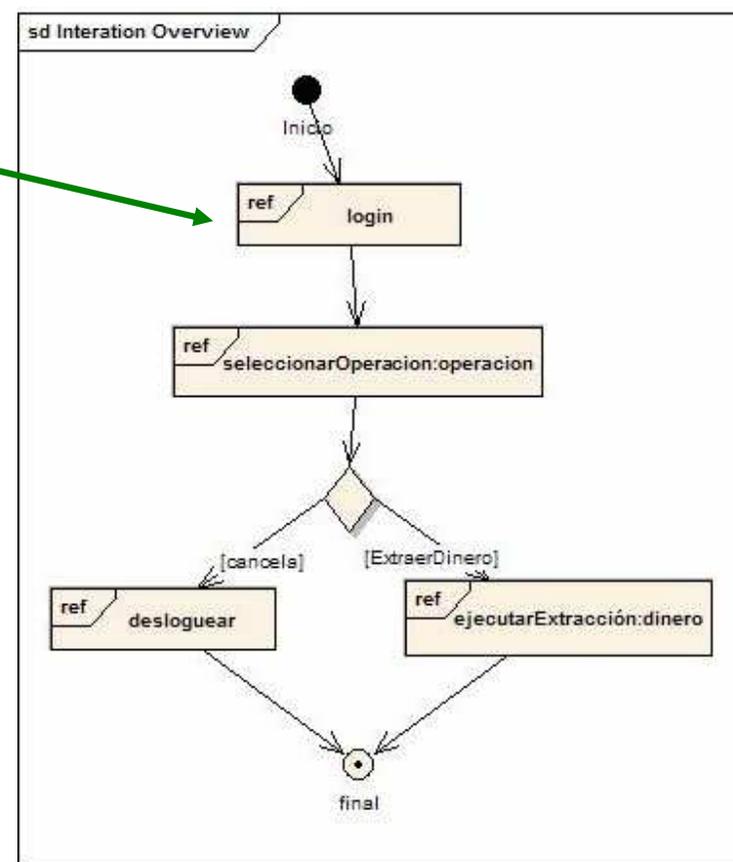
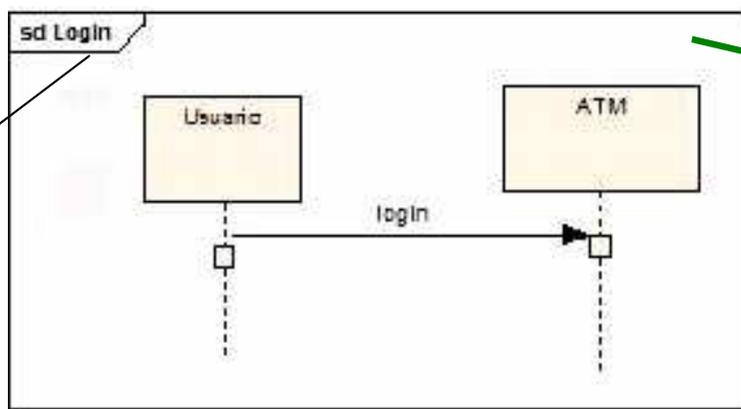




Diagramas de Interacción– Control Estructurado

- Los fragmentos de interacción permiten reutilizar interacciones en otros diagramas de interacción (incluso diagramas completos)

Nombre de la interacción



Definición de la interacción “Login” como un diagrama de secuencia (izquierda) y su reutilización en un diagrama de revisión de interacciones (derecha).



Diagramas de Interacción – Diagramas 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 a través de los enlaces.

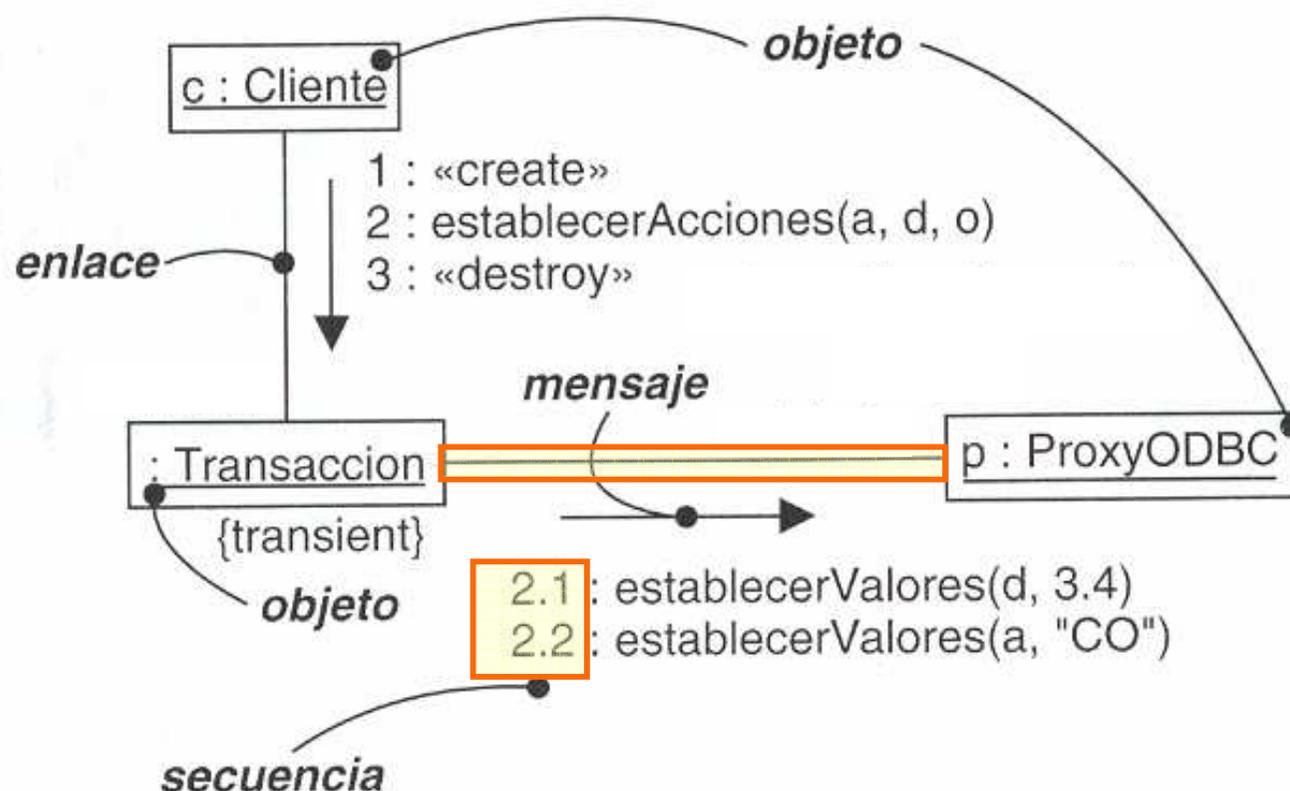


- En versiones anteriores a UML 2 se llamaban de colaboración.



Diagramas de Interacción – Diagramas de Comunicación

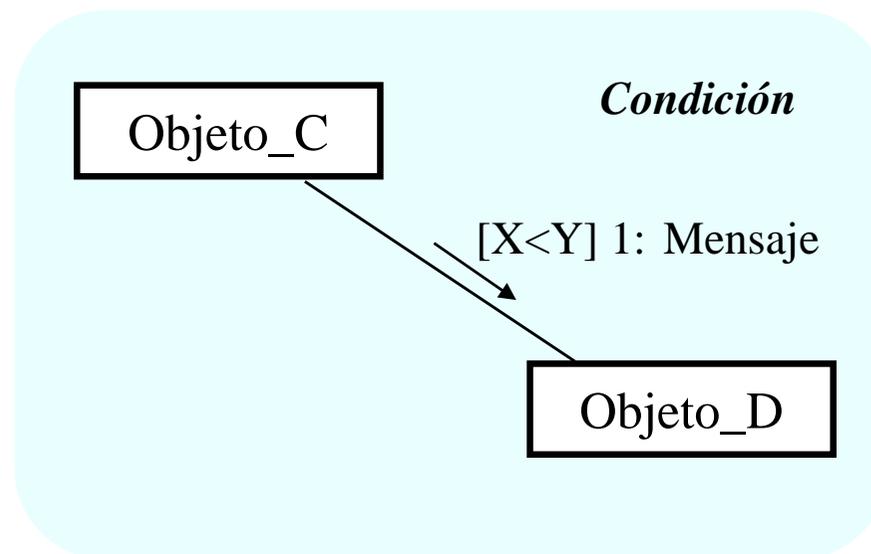
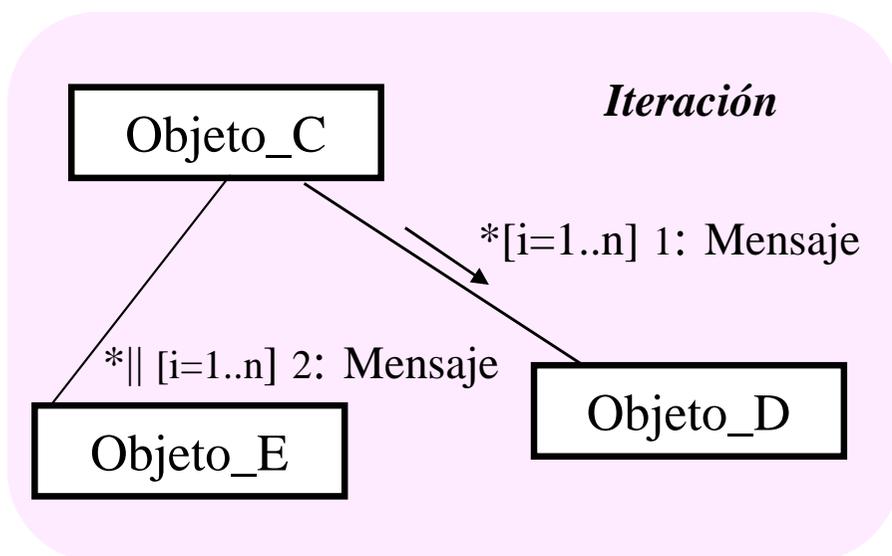
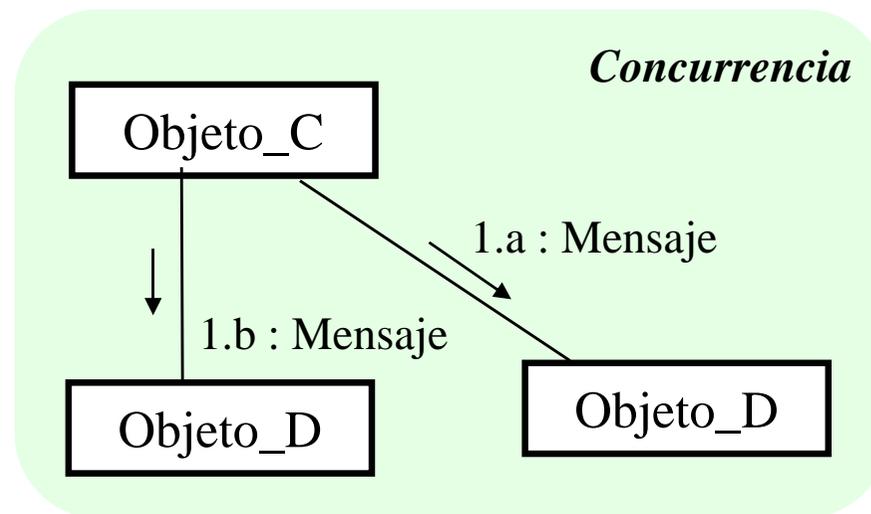
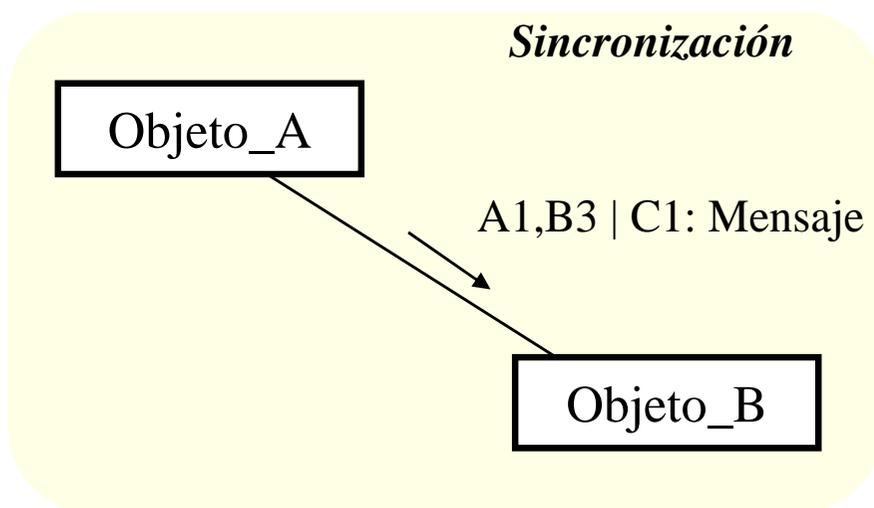
- Tienen dos características que los distinguen de los diagramas de secuencia:
 - **Camino**: Indica cómo se enlaza un objeto a otro.
 - **Número de Secuencia**: Indica la ordenación temporal de los mensajes.





Diagramas de Interacción – Diagramas de Comunicación

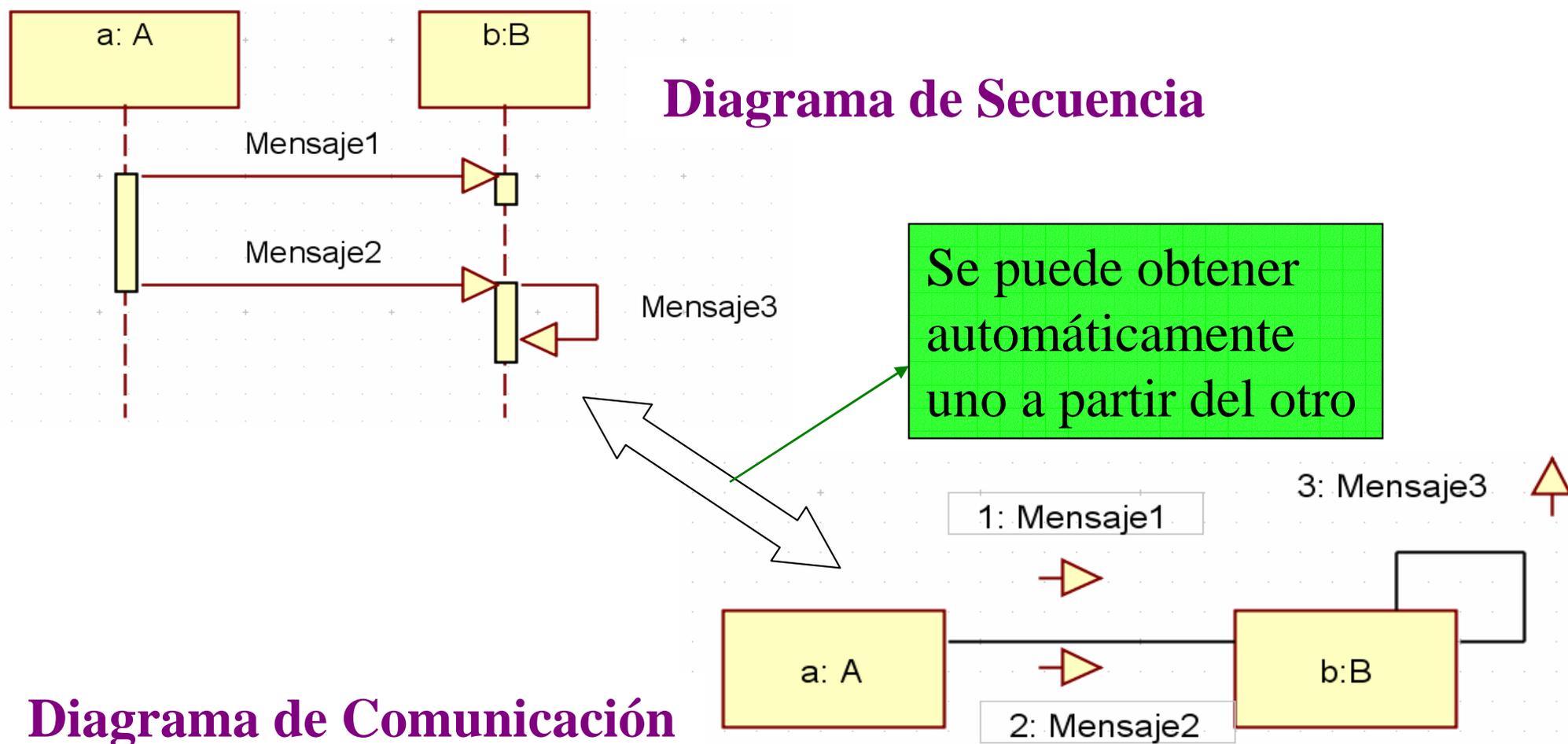
- Estructuras de control en diagramas de comunicación:





Diagramas de Interacción – Equivalencia

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





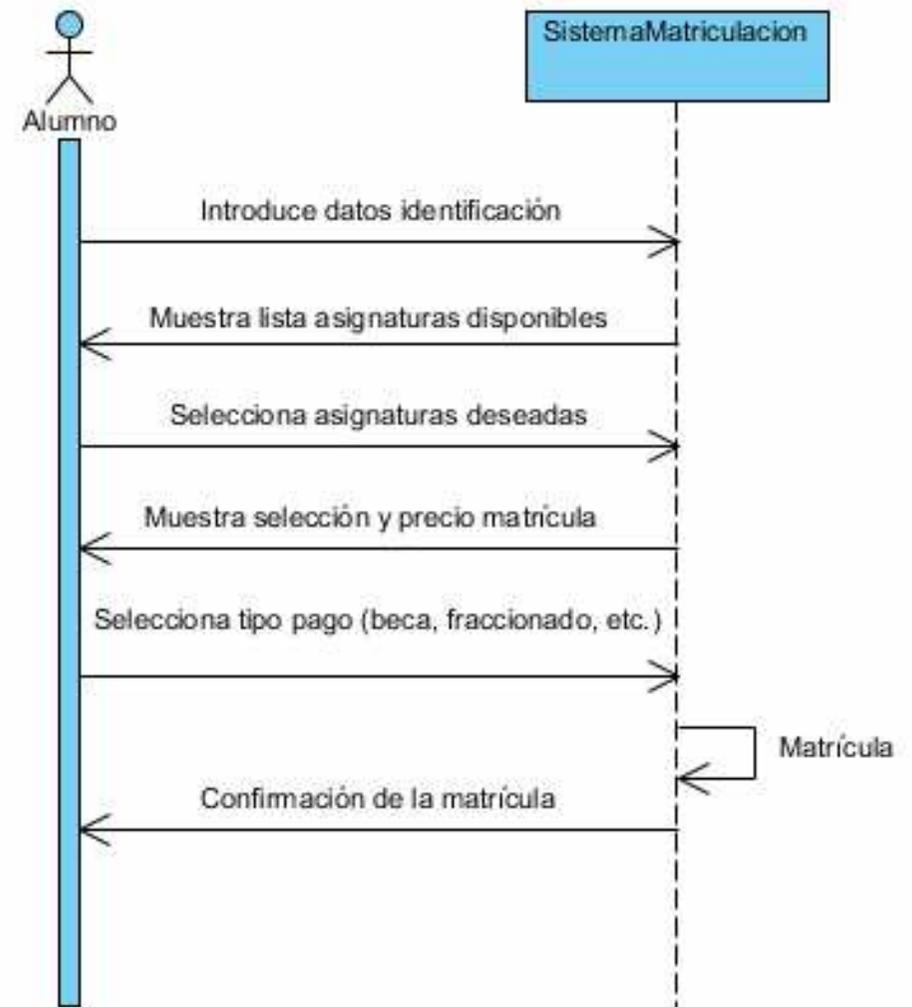
Modelado – Requisitos vs Diseño

- Los diagramas de interacción se utilizan de manera diferente, según 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**.
 - Alto nivel de abstracción
 - **DISEÑO**: Representar de forma precisa las interacciones entre los objetos que forman el sistema.
 - Durante la **operación del sistema**, los objetos del sistema interactúan entre sí para implementar la funcionalidad del sistema.
 - Bajo nivel de abstracción, mucho detalle.



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.
 - Las flechas no corresponden aún a envíos de mensajes en el sentido de lenguajes de programación.

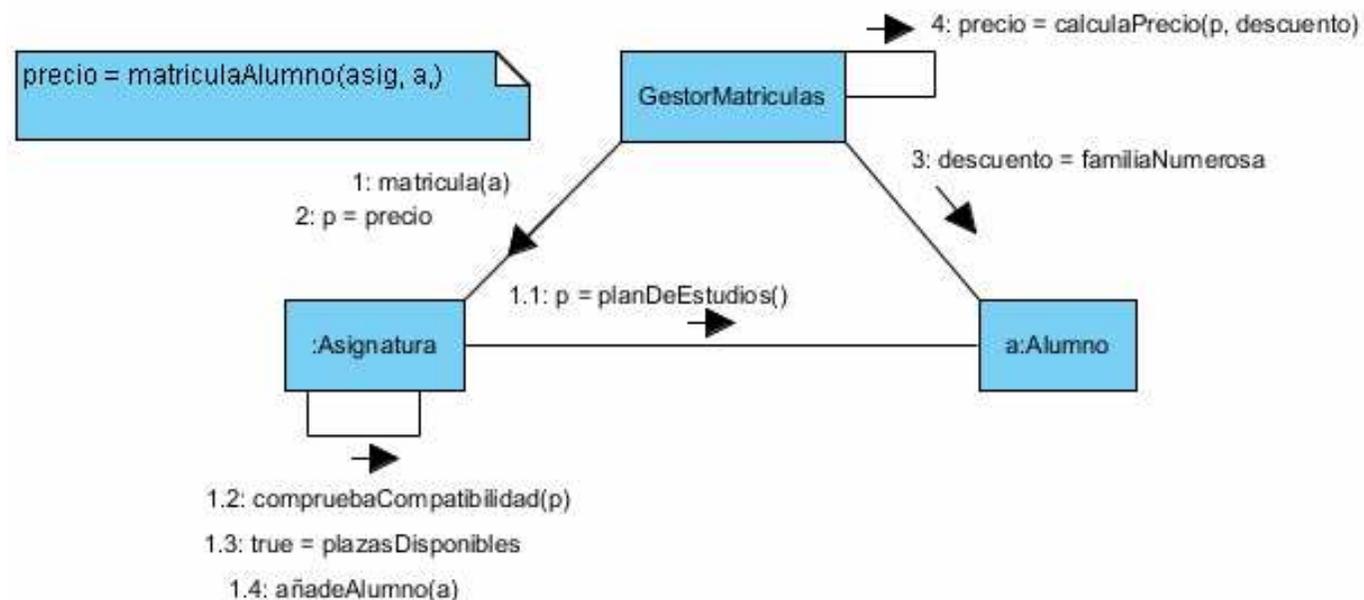




Modelado – Requisitos vs Diseño

- Durante la etapa de Diseño

- El diagrama de interacción es una **representación precisa de las interacciones entre los objetos**.
 - Realización de casos de uso por parte de los objetos que forman la estructura estática del sistema.
- 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.





Ejercicio de diagramas de interacción

Identificador: Realiza_pedido

Actor que lo inician: Cliente

Precondiciones: Un cliente registrado en el sistema ha accedido correctamente al sistema.

Secuencia de eventos de flujo:

1. El cliente introduce su nombre y dirección.
2. El cliente introduce el código del producto que desea incluir en el pedido.
4. El sistema aporta la descripción y el precio del producto.
5. El sistema almacena temporalmente el pedido.
6. El cliente introduce la información de la tarjeta de pago.
7. El cliente pulsa el control Ejecuta.
8. El sistema requiere confirmación del banco.
9. El pago es confirmado, se acepta el pedido, se almacena en la base de datos, y se le asigna un ID que se retorna al cliente

Postcondiciones: Si el pedido no ha sido cancelado, es registrado en el sistema y confirmado al cliente.



Ejercicio de diagramas de interacción

Identificador: Realiza_pedido

Actor que lo inician: Cliente

Precondiciones: Un cliente registrado en el sistema ha accedido correctamente al sistema.

Secuencia de eventos de flujo:

1. El cliente introduce su nombre y dirección.
2. Si el cliente introduce el ZIP, el sistema introduce la ciudad y región.
3. El cliente introduce los códigos de los productos que desea incluir en el pedido.
4. Por cada código de producto introducido:
 - 4.a El sistema aporta la descripción y el precio del producto.
 - 4.b El sistema suma el precio del producto en el total del pedido.
5. El sistema almacena temporalmente la lista de productos incluidos en el pedido.
6. El cliente introduce la información de la tarjeta de pago.
7. El cliente pulsa el control Ejecuta.
8. El sistema requiere confirmación del banco. Si la información es incorrecta, el sistema requiere su corrección al cliente.
9. Cuando el pago es confirmado, se acepta el pedido, se almacena y se le asigna un ID que se retorna al cliente

Postcondiciones: Si el pedido no ha sido cancelado, es registrado en el sistema y confirmado al cliente.



Ejercicio de diagramas de interacción

