

UT04. DISEÑO ORIENTADO A OBJETOS. ELABORACIÓN DE DIAGRAMAS DE COMPORTAMIENTO.

Entornos de Desarrollo

1 DAW – C.I.F.P. Carlos III - Cartagena

Índice

- 1.- Introducción a la orientación a objetos
- 2.- Conceptos de orientación a objetos
 - 2.1.- Ventajas de la orientación a objetos.
 - 2.2.- Clases, atributos y métodos.
 - 2.3.- Visibilidad
 - 2.4.- Objetos. Instanciación.
- 3.- UML
 - 3.1.- Tipos de diagramas UML.
 - 3.2.- Herramientas para la elaboración de diagramas UML.
 - 3.3.- Diagramas de clases.
 - 3.4.- Relaciones entre clases.
 - 3.5.- Paso de los requisitos de un sistema al diagrama de clases.
 - 3.6.- Generación de código a partir del diagrama de clases.
 - 3.7.- Generación de la documentación.
- 4.- Ingeniería inversa.

1. Diagramas de comportamiento.

Introducción

- Diagrama de clases ➔ información estática
- Diagramas de comportamiento ➔ comportamiento dinámico
 - Diagramas de casos de uso. Use Case Diagram
 - Diagramas de actividad.
 - Diagramas de máquinas de estado.
 - Diagramas de interacción.
 - ➔ Diagramas de secuencia.
 - ➔ Diagramas de comunicación (EX-**colaboración**).
 - ➔ Diagramas de interacción.
 - ➔ Diagramas de tiempo

2. Diagramas de casos de uso

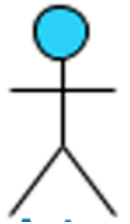
Diagramas de casos de uso

- **Resuelve** la falta de comunicación entre el equipo de desarrollo y el equipo que necesita de una solución software.
- **Determina QUÉ** puede hacer cada tipo diferente de usuario con el sistema.
- **Documenta** el comportamiento de un sistema desde el punto de vista del usuario ➔ determinan los requisitos funcionales del sistema.
- Representación gráfica de los Requisitos Funcionales del Sistema:
 - Casos de uso-Elipses
 - Actores-Monigotes
- Facilidad de interpretación en la comunicación con el cliente

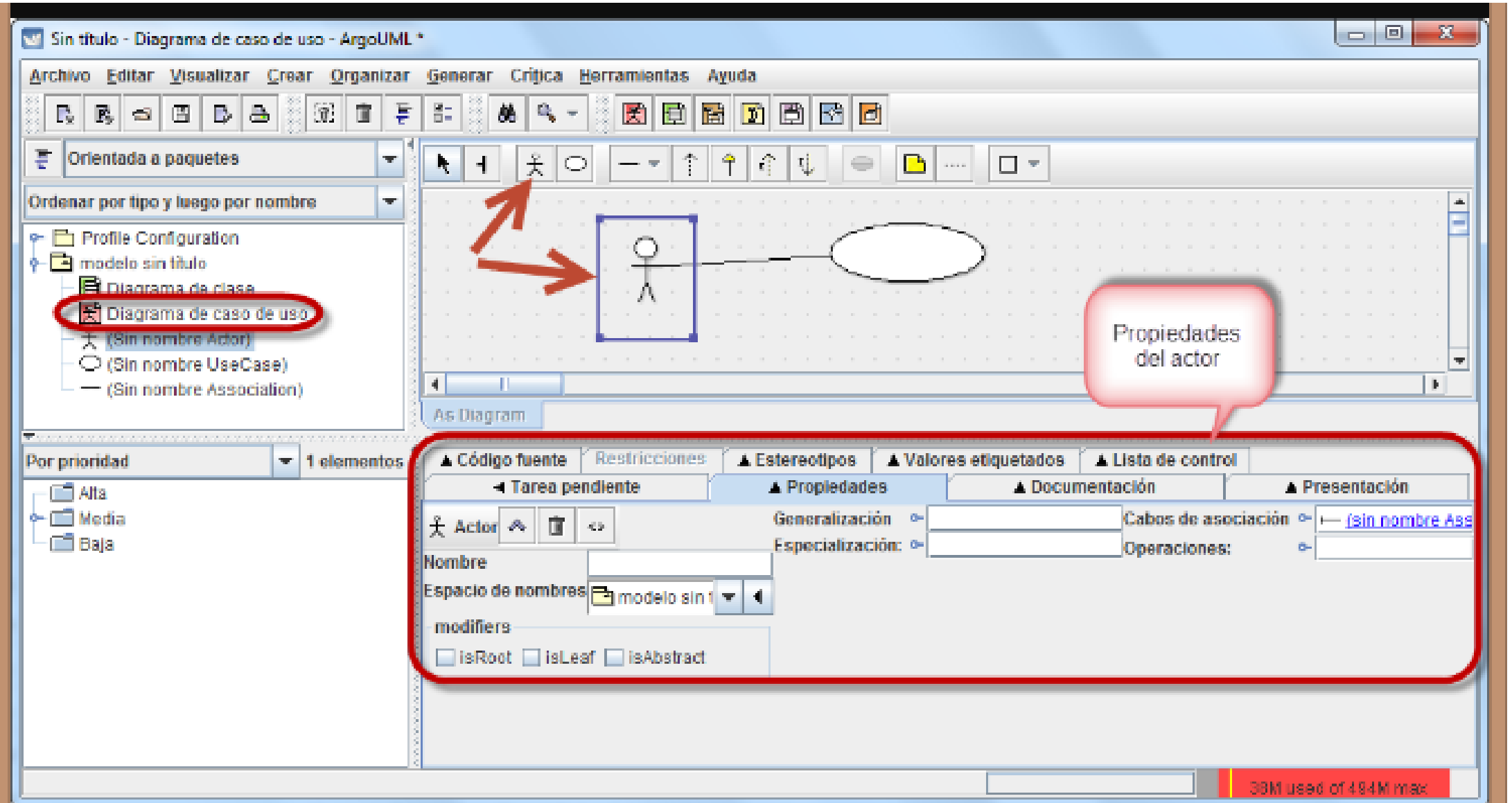
Diagramas de casos de uso

- Se enmarcan en el proceso de análisis➡
 - Detalle de la funcionalidad
 - De los diagramas de casos de uso se desprenden otros diagramas.
- Se usan en la fase de prueba, para verificar que el sistema cumple con los **requisitos funcionales**

Diagramas de casos de uso. Actores



- Representan un **tipo de usuario** del sistema.
- **Usuario** = cualquier cosa externa que interactúa con el sistema (humano, otro sistema informático o unidades organizativas o empresas).
- Rol vs usuario → un actor por cada rol que interpreta un usuario
- Tipos.
 - **Primarios:** interaccionan con el sistema para explotar su funcionalidad. Trabajan directa y frecuentemente con el software.
 - **Secundarios:** soporte del sistema para que los primarios puedan trabajar. Son precisos para alcanzar algún objetivo.
 - **Iniciadores:** no interactúan con el sistema pero desencadenan el trabajo de otro actor.
- Casos de uso que no sean iniciados por ningún usuario, o algún otro elemento software, en ese caso se puede crear un actor "**Tiempo**" o "**Sistema**".



Casos de uso

- Secuencia de **acciones**, incluyendo variantes, que el sistema puede llevar a cabo, y que producen un resultado observable de valor para un actor concreto.
- El conjunto de casos de uso forma el "**comportamiento requerido**" de un sistema.
- **Objetivo.** Descripción que de cada caso se debe realizar.

Casos de uso.

- Datos-Documentación:
 - **Nombre:** nombre del caso de uso.
 - **Actores:** aquellos que interactúan con el sistema a través del caso de uso.
 - **Propósito:** breve descripción de lo que se espera que haga.
 - **Precondiciones:** aquellas que deben cumplirse para que pueda llevarse a cabo el caso de uso.
 - **Flujo normal:** flujo normal de eventos que deben cumplirse para ejecutar el caso de uso exitosamente, desde el punto de vista del actor que participa y del sistema.
 - **Flujo alternativo:** flujo de eventos que se llevan a cabo cuando se producen casos inesperados o poco frecuentes. No se deben incluir aquí errores como escribir un tipo de dato incorrecto o la omisión de un parámetro necesario.
 - **Postcondiciones:** las que se cumplen una vez que se ha realizado el caso de uso.

Sin título - Diagrama de caso de uso - ArgoUML *

Archivo Editar Visualizar Crear Organizar Generar Crítica Herramientas Ayuda

Orientada a paquetes

Ordenar por tipo y luego por nombre

- Profile Configuration
- modelo sin título
 - Diagrama de clase
 - Diagrama de caso de uso
 - (Sin nombre Actor)
 - (Sin nombre UseCase)**
 - (Sin nombre Association)

Por prioridad 1 elementos

- Alta
- Media
- Baja

As Diagram

Diagrama: Un actor conectado a un caso de uso (oval). Red arrows point from the UseCase icon in the toolbar to the UseCase element in the diagram.

Propiedades del caso de uso

▲ Código fuente	Restricciones	▲ Estereotipos	▲ Valores etiquetados	▲ Lista de control
▲ Tarea pendiente		▲ Propiedades		
○ UseCase		Generalización: <input type="text"/>		
Nombre: <input type="text"/>		Especialización: <input type="text"/>		
Espacio de nombres: <input type="text"/> sin título		Extensor: <input type="text"/>		
modifiers: <input type="checkbox"/> isRoot <input type="checkbox"/> isLeaf <input type="checkbox"/> isAbstract		Incluye: <input type="text"/>		
Dependencias del cliente: <input type="text"/>		Atributos: <input type="text"/>		
Dependencias del surtidor: <input type="text"/>		Operaciones: <input type="text"/>		
		Cabos de asociación: <input type="text"/> (sin nombre Asso)		
		Puntos de extensión: <input type="text"/>		

4UM used of 494M max

Diagramas de casos de uso. Relaciones.

- Representan la asociación entre y casos de uso.
- Tipos de relaciones entre elementos:
 - **Asociación:** representa la relación entre el actor que lo inicia y el caso de uso.
 - **Inclusión:** se utiliza cuando queremos dividir una tarea de mayor envergadura en otras más sencillas, que son utilizadas por la primera. Representa una relación de uso, y son muy útiles cuando es necesario reutilizar tareas.
 - **Extensión:** se utiliza para representar relaciones entre un caso de uso que requiere la ejecución de otro en determinadas circunstancias.
 - **Generalización:** se utiliza para representar relaciones de herencia entre casos de uso o actores.

Estereotipo

- **Estereotipo:** elemento de texto que al ser aplicado a otro elemento define su categoría.
- Es uno de los **mecanismos de extensión** del lenguaje toda vez que permite cambiar o complementar la semántica de cualquier elemento.
- Uno de estos mecanismos de extensión de los elementos del lenguaje UML, quizás el más usado, son los **estereotipos**; pequeñas etiquetas que aplicadas a los elementos o relaciones de un diagrama indican significado adicional.
- La especificación UML estandariza un conjunto de estereotipos que se pueden aplicar a los elementos de un modelo. En general, cualquier diseñador puede crear conjuntos de estereotipos.
- La notación de un estereotipo es una cadena de texto encerrada entre los símbolos de comillas francesas (« »).

Diagramas de casos de uso.

Relaciones. Interacción o asociación

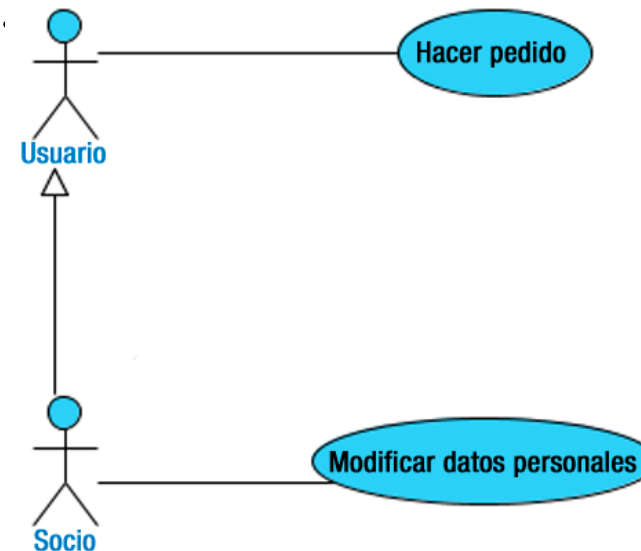
- Asociación entre un actor y un caso de uso.
- Se representa mediante un línea continua que une un actor con un caso de uso.
- Por ejemplo, un usuario de un sistema de venta por Internet puede hacer un pedido.

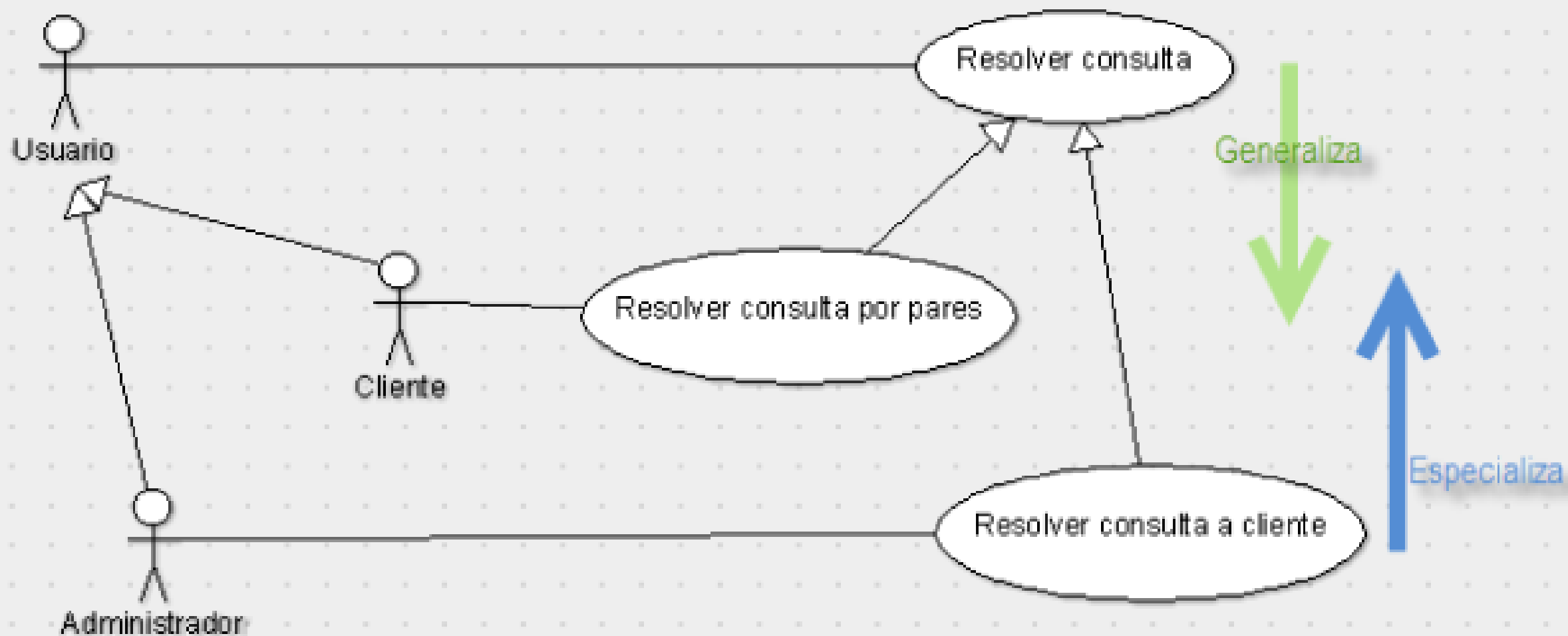


Diagramas de casos de uso.

Relaciones. Generalización

- Existen casos de uso con comportamientos semejantes a otros que los modifican o completan de alguna manera.
- El **caso base** se define de forma abstracta y los hijos heredan sus características añadiendo sus propios pasos o modificando alguno.
- Normalmente la **herencia** se utiliza menos en diagramas de casos de uso que en diagramas de clases.
- Por ejemplo, el usuario del sistema de venta por Internet puede a su vez darse de alta en la página web para que tengan sus datos registrados a la hora de hacer el pedido, en este caso el usuario es la generalización del socio. Ambos actores pueden hacer un pedido, pero solo el socio puede modificar sus datos en el sistema.

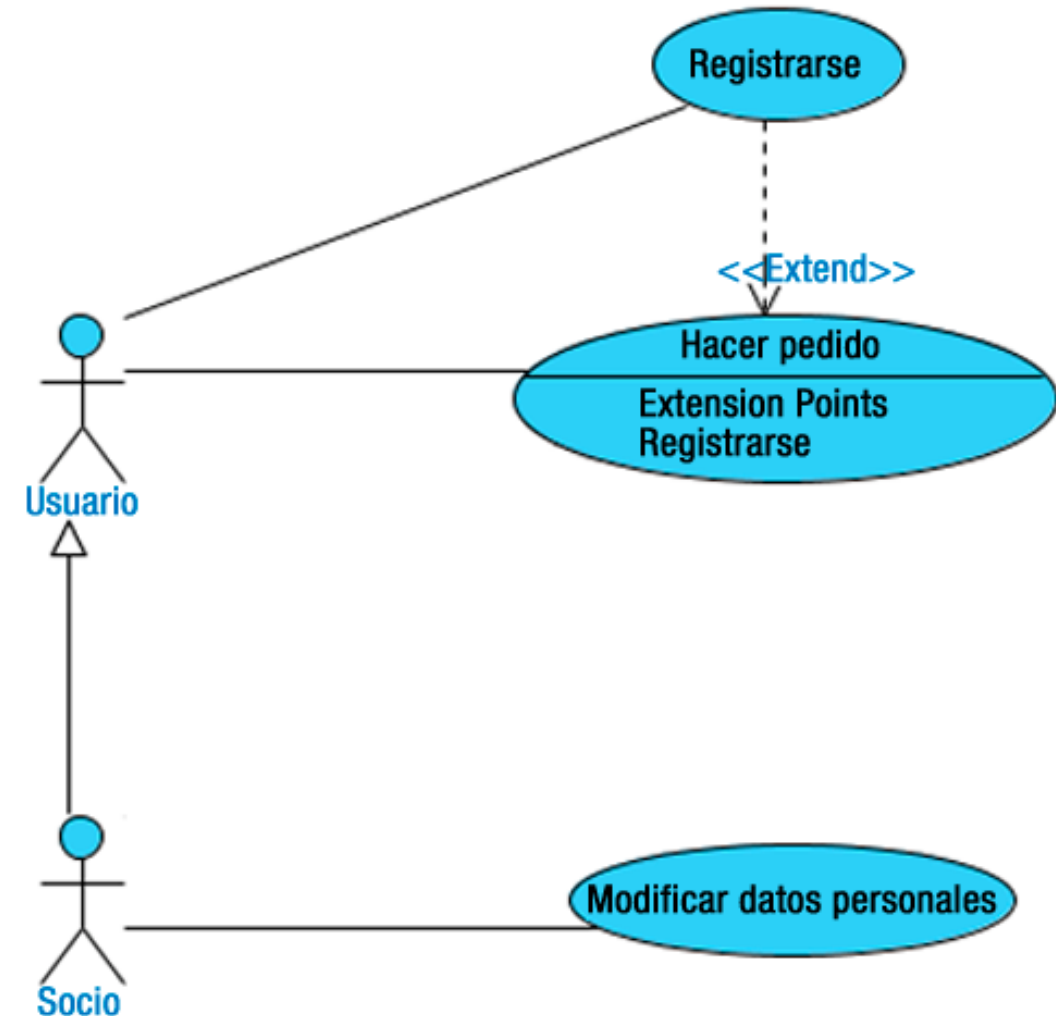




Diagramas de casos de uso.

Relaciones. Extensión

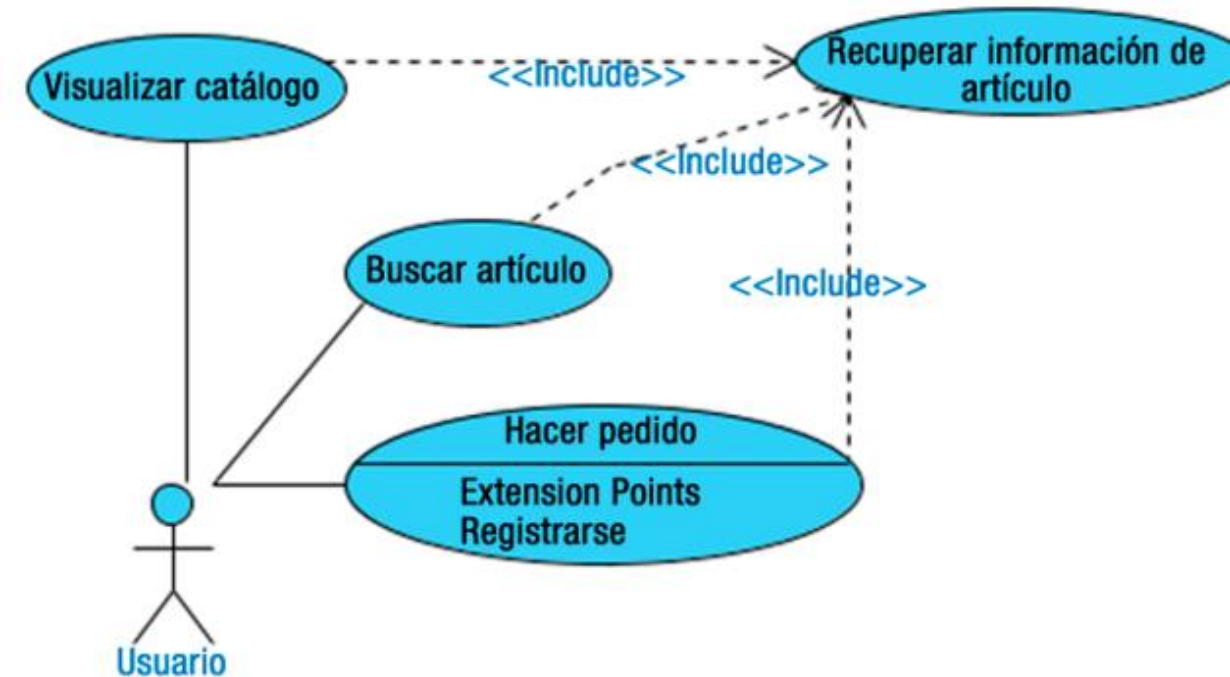
- Relación entre dos casos de uso de tipo "**extends**". Para especificar que el comportamiento de un caso de uso es diferente dependiendo de ciertas circunstancias.
- Si existe una parte del caso de uso que se ejecuta sólo en determinadas ocasiones, pero no es imprescindible para su completa ejecución.
- **Ejemplo.-** cuando un usuario hace un pedido si no es socio se le ofrece la posibilidad de darse de alta en el sistema en ese momento, pero puede realizar el pedido aunque no lo sea.



Diagramas de casos de uso.

Relaciones. Inclusión

- Relación entre dos casos de uso de tipo "**include**". Cuando la ejecución del caso de uso incluido se da en la rutina normal del caso que lo incluye.
- **Uso.** Cuando se desea especificar algún comportamiento común en dos o más casos de uso.
- **Error frecuente.** Hacer subdivisión de funciones.
- **Ejemplo:** para hacer un pedido se debe buscar la información de los artículos para obtener el precio (proceso que forma parte del caso de uso y de otros)



Diagramas de casos de uso.

Relaciones. Inclusión

- **Ventajas.**

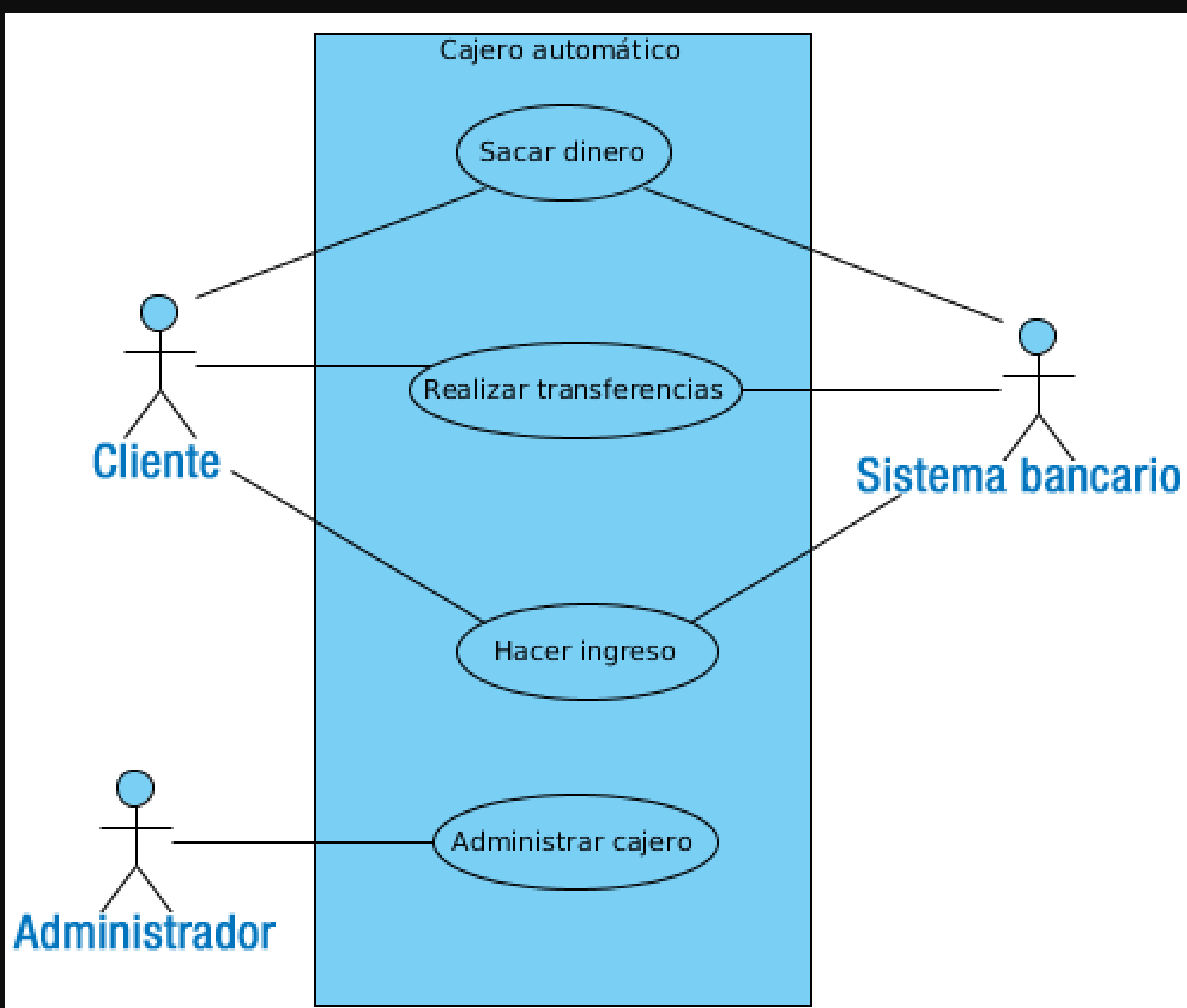
- Las descripciones de los casos de uso son más cortas y se entienden mejor.
- La identificación de funcionalidad común puede ayudar a descubrir el posible uso de componentes ya existentes en la implementación.

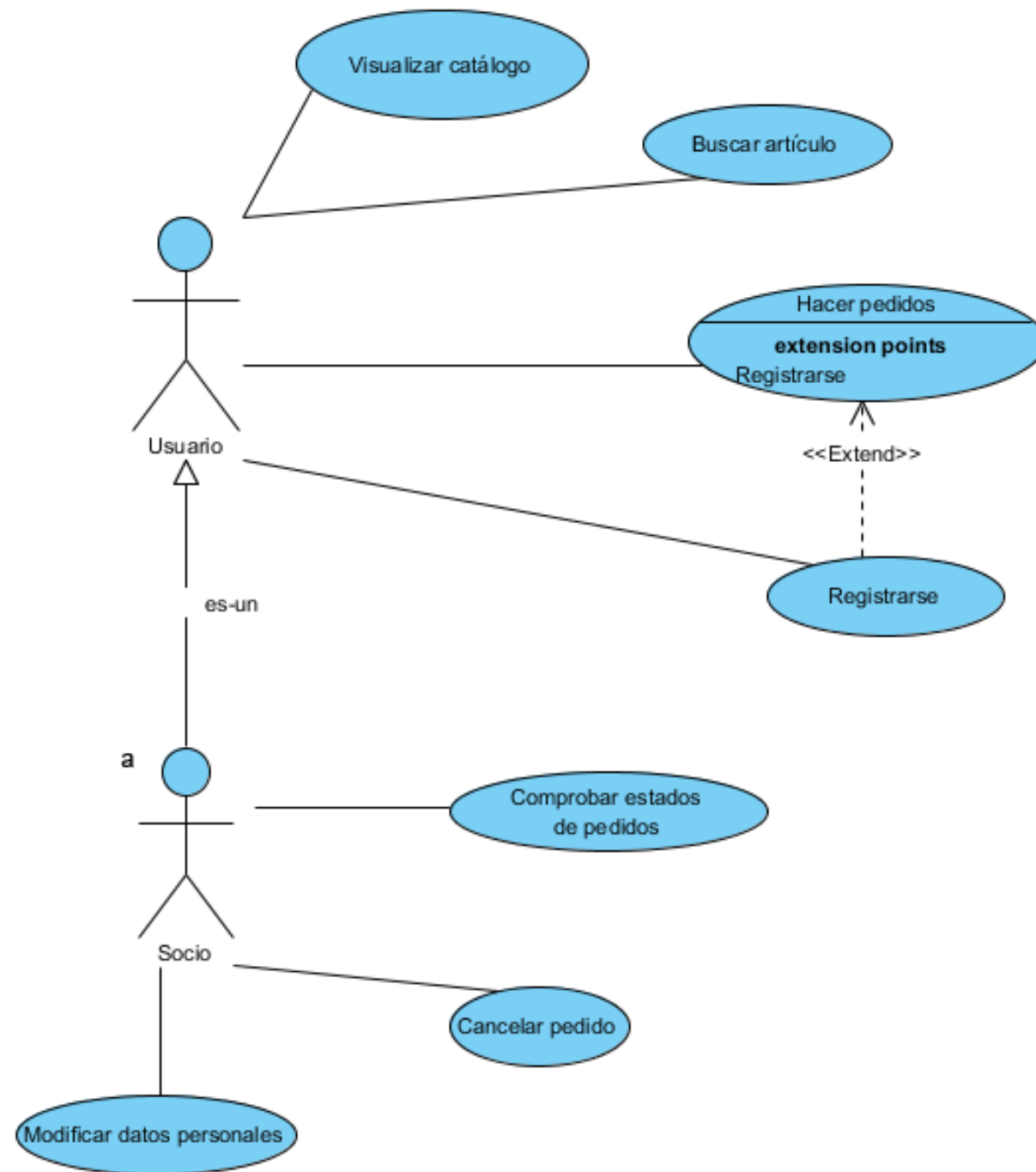
- **Desventajas.**

- La inclusión de estas relaciones hace que los diagramas sean más difíciles de leer, sobre todo para los clientes.

Diagramas de casos de uso. Elaboración de casos de uso

- Descripción lo más detallada posible del problema a resolver
- Detectar quien interactúa con el sistema → Actores
- Tareas que realizan los actores → casos de uso más genéricos.
- Refinar el diagrama: casos de uso más generales, casos relacionados por inclusión, extensión y generalización.
- Al diagrama generado se le denomina **diagrama frontera**.
 - Incluye todos los casos de uso genéricos del sistema, que podrán ser desglosados después en nuevos diagramas de casos de uso que los describan si es necesario.
 - Se especifica enmarcando los casos de uso en un recuadro, que deja a los actores fuera.





Diagramas de casos de uso. Escenarios

- Un **escenario** es una ejecución particular de un caso de uso que se describe como una secuencia de eventos.
- Un caso de uso es una generalización de un escenario.
- Los escenarios pueden y deben posteriormente documentarse mediante diagramas de secuencia.
- Ejemplo.
 - Caso de uso: <Hacer pedido>
 - Diferentes escenarios

Diagramas de casos de uso. Escenarios

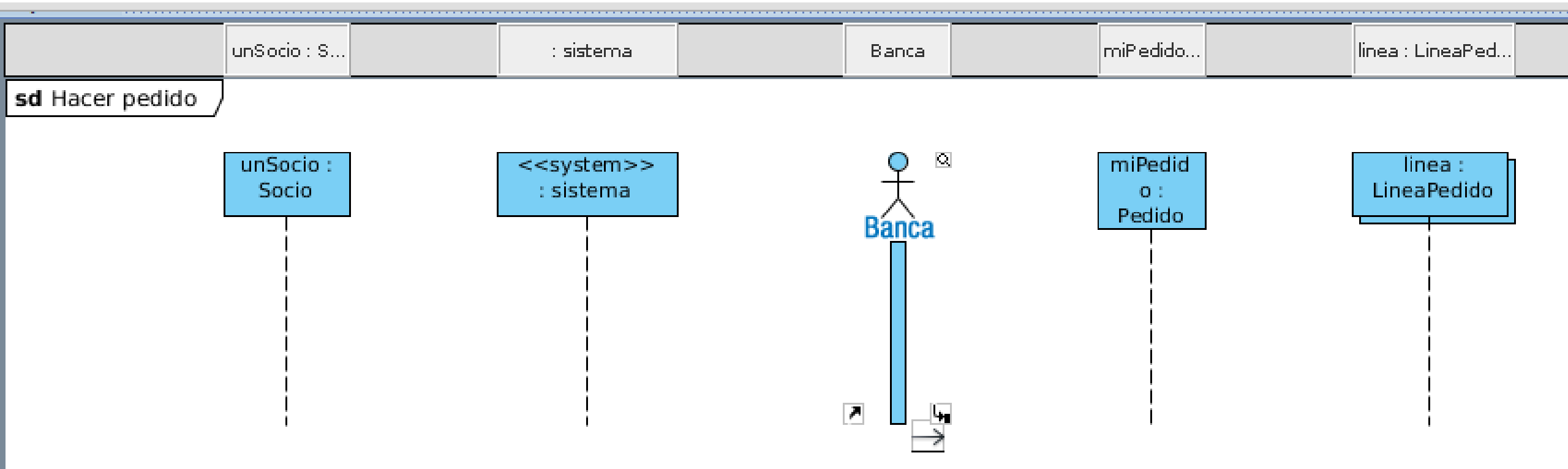
- Realizar pedido de unos zapatos y unas botas.
 - El usuario inicia el pedido.
 - Se crea el pedido en estado "en construcción".
 - Se selecciona un par de zapatos "Lucía" de piel negros, del número 39.
 - Se selecciona la cantidad 1.
 - Se recupera la información de los zapatos y se modifica la cantidad a pagar sumándole 45 €.
 - Se selecciona un par de botas "Aymara" de ante marrón del número 40.
 - Se selecciona la cantidad 1.
 - Se recupera la información de las botas y se modifica la cantidad a pagar sumándole 135 €.
 - El usuario acepta el pedido.
 - Se comprueba que el usuario es, efectivamente socio.
 - Se comprueban los datos bancarios, que son correctos.
 - Se calcula el total a pagar añadiendo los gastos de envío.
 - Se realiza el pago a través de una entidad externa.
 - Se genera un pedido para el usuario con los dos zapatos que ha comprado, con el estado "pendiente".

3. Diagramas de secuencia

Diagramas de secuencia

- **Uso:** formalizar la **descripción de un escenario** o conjunto de ellos representando qué **mensajes** fluyen en el sistema así como quien los envía y quien los recibe.
- **Objetos y actores** → rectángulos distribuidos horizontalmente en la zona superior del diagrama.
- **Línea temporal vertical** (una por cada actor) con los diferentes mensajes que se pasan entre ellos.
- Completan a los diagramas de casos de uso.

Diagramas de secuencia



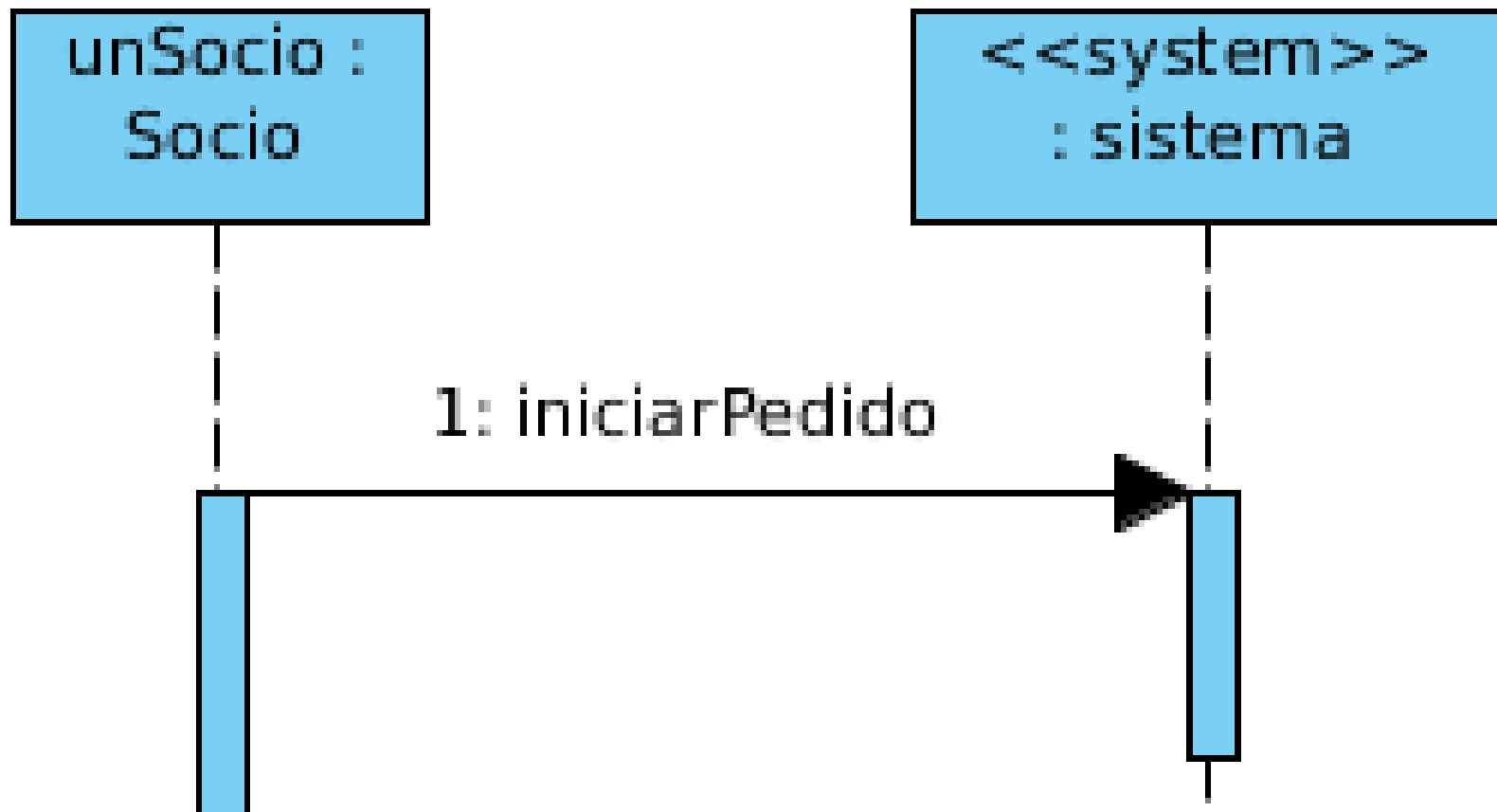
Diagramas de secuencia

- **Representación de objetos y línea de vida.**
- **Entidades** que participan = rectángulos distribuidos horizontalmente.
- **Línea de vida.** Sale de cada rectángulo o entidad una línea de puntos (paso del tiempo).
<nombreObjeto>:<nombreClase>
- Encabezada por instancias: sistema o un actor.
- Usar el sistema para representar solicitudes al mismo, como por ejemplo pulsar un botón para abrir una ventana o una llamada a una subrutina.
- Objeto que puede tener varias instancias: rectángulo sobre otro (el caso de las líneas del pedido que pueden ser varias).

Diagramas de secuencia

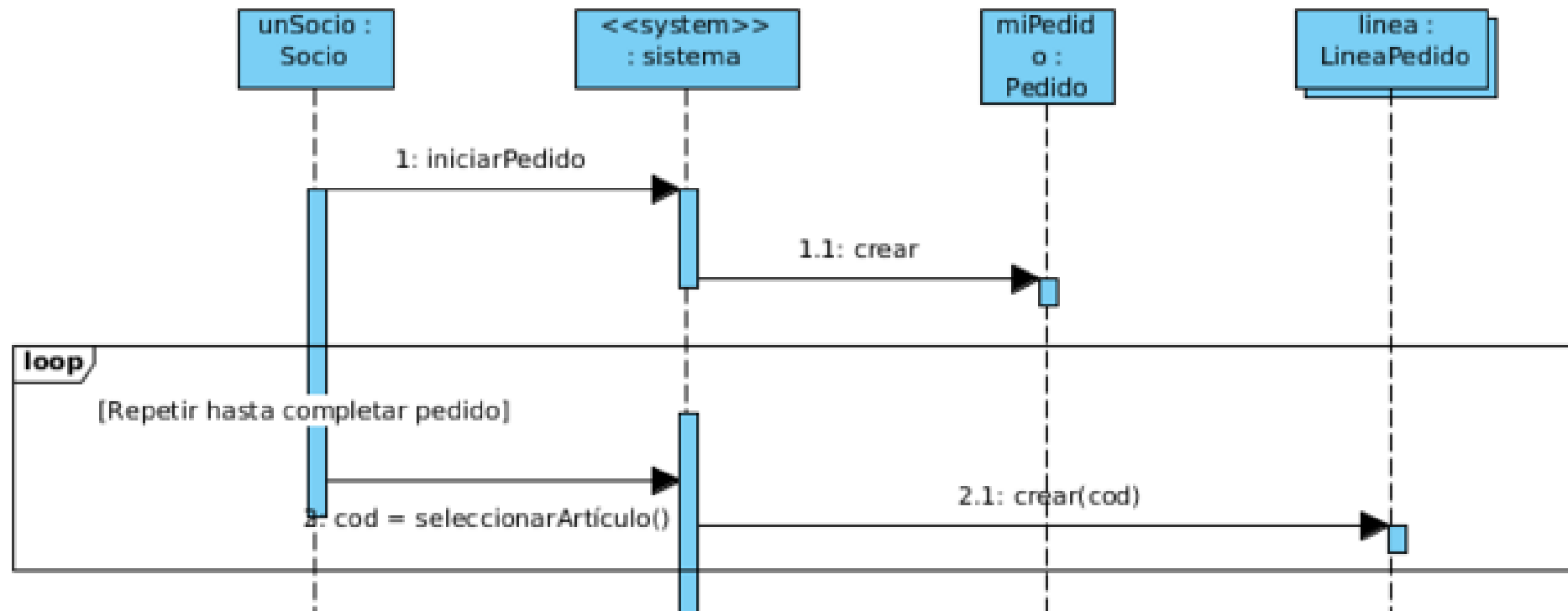
- **Invocación de métodos.**

- Los **mensajes** -invocación de métodos- flechas horizontales de una línea de vida a otra, indicando con la flecha la dirección del mensaje, los extremos de cada mensaje se conectan con una línea de vida que conecta a las entidades al principio del diagrama.
- Pudiendo ser ambos el mismo objeto
- Orden viene determinado por su posición vertical con un número de secuencia.



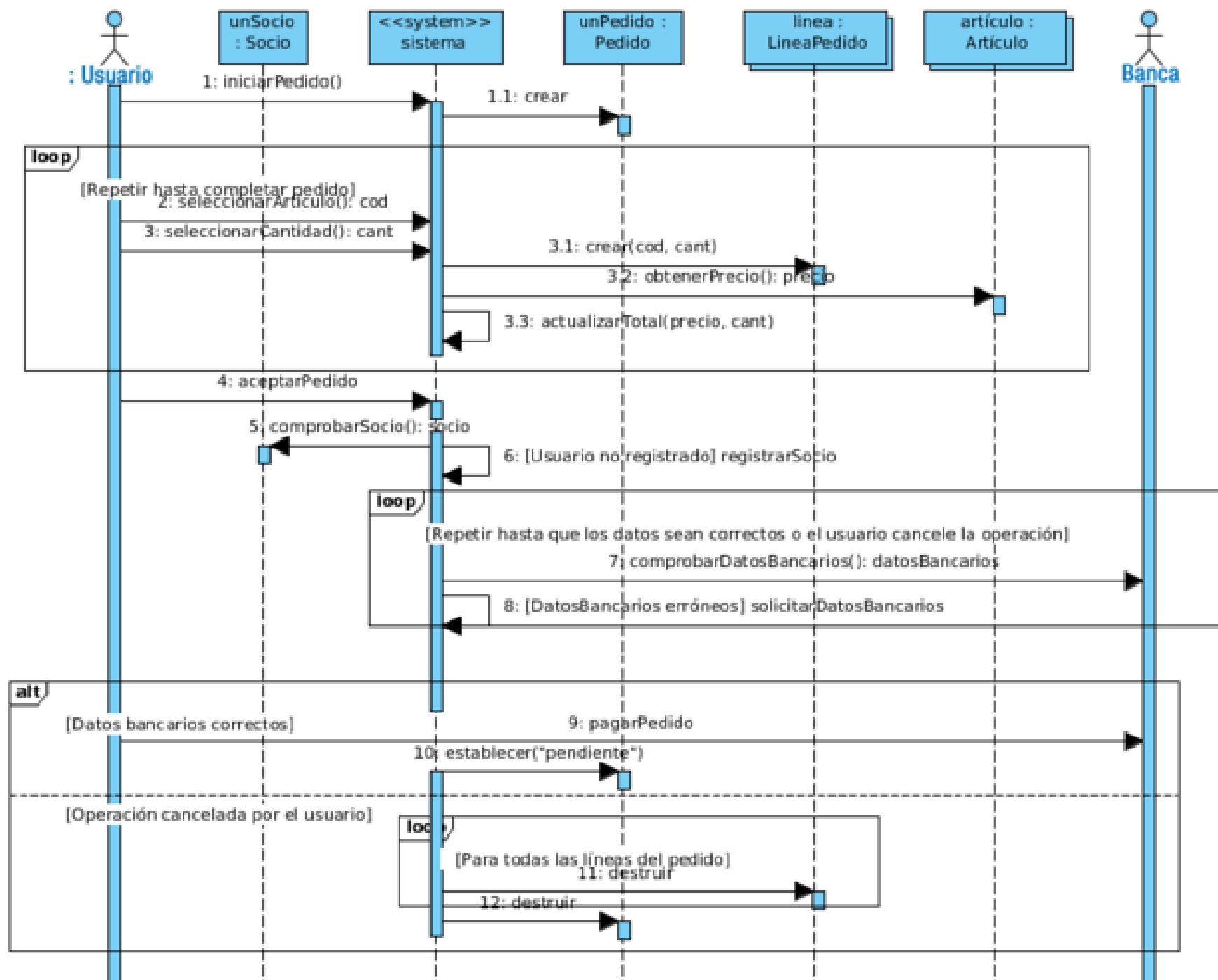
- **Iteraciones y condicionales.**

- Bucles = marcos y la condición de parada. También se pueden representar flujos de mensajes condicionales en función de un valor determinado.
- Se puede añadir etiquetas y notas en el margen izquierdo que aclare la operación que se está realizando.

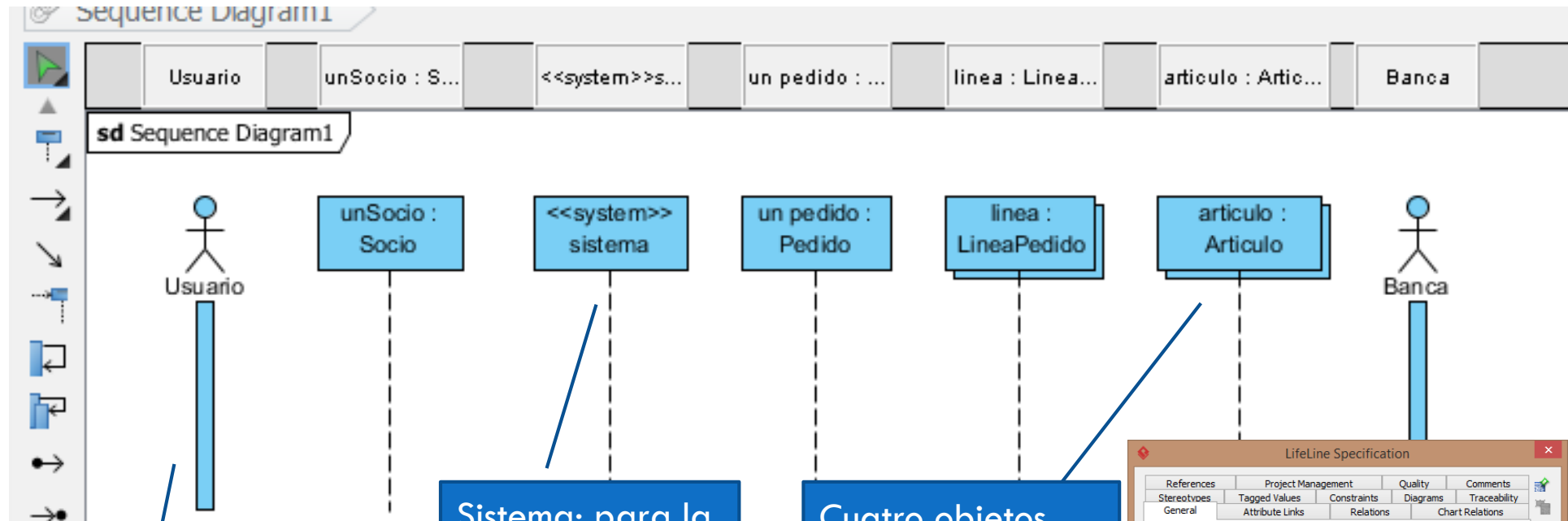


Elaboración de un diagrama de secuencia.

- Ejemplo.- **Hacer pedido.**
 - Secuencia de operaciones que se llevarán a cabo entre los diferentes objetos que intervienen en el caso de uso.



Incluir entidades



Dos actores:
Usuario: actor principal
Banca: actor secundario que actúa por el usuario comprobando los datos de la tarjeta

Sistema: para la interacción con interfaz gráfica.
En Open Specification>> pestaña Stereotypes>> seleccionar system

Cuatro objetos mas el sistema.
Cuando hay mas de un objeto implicado hay que marcar la casilla Multi-object de Open Specification

LifeLine Specification

References

Project Management

Quality

Comments

Stereotypes

Tagged Values

Constraints

Diagrams

Traceability

General

Attribute Links

Relations

Chart Relations

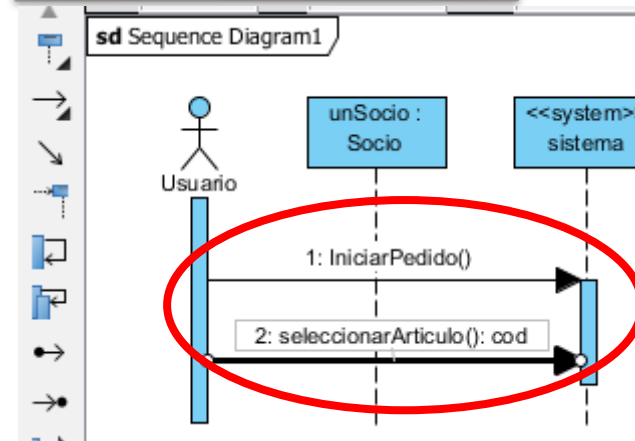
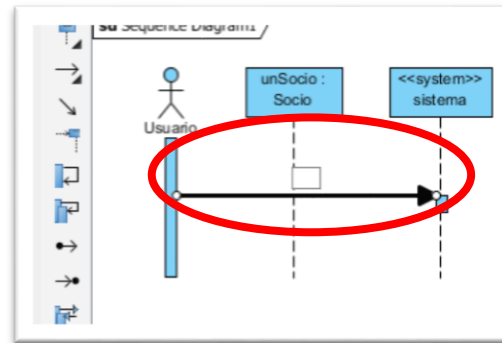
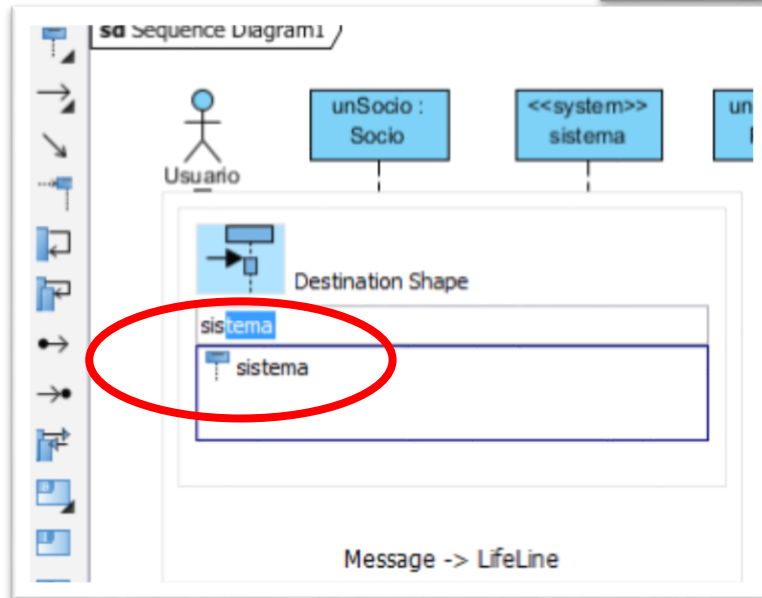
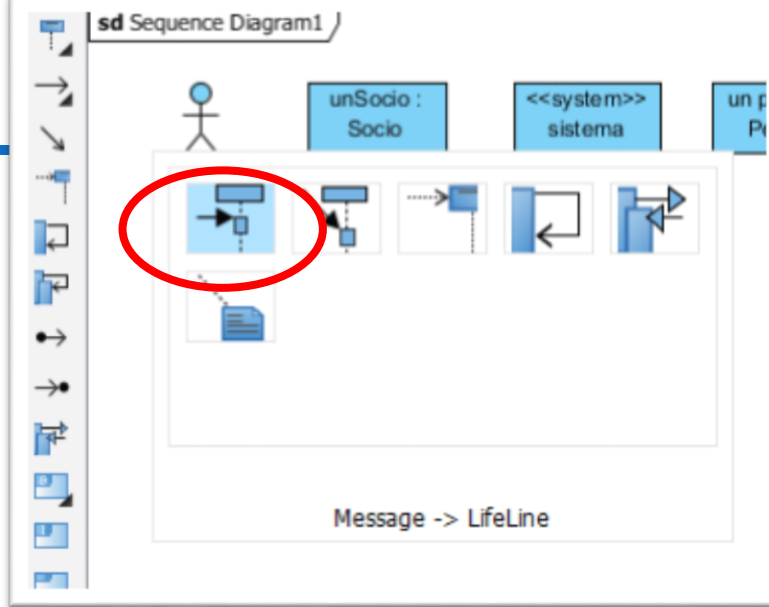
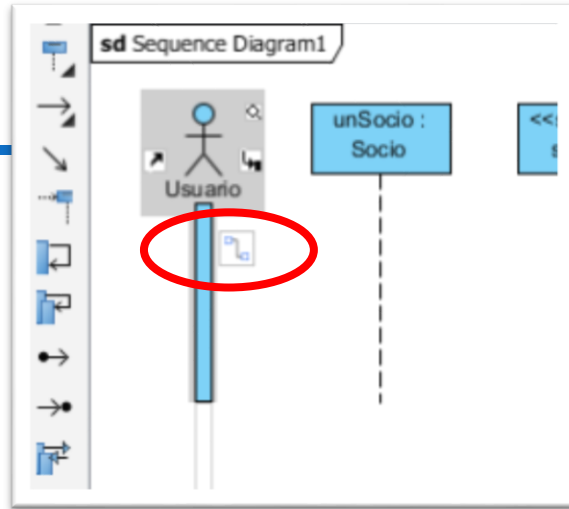
Name: articulo

Base classifier: Articulo

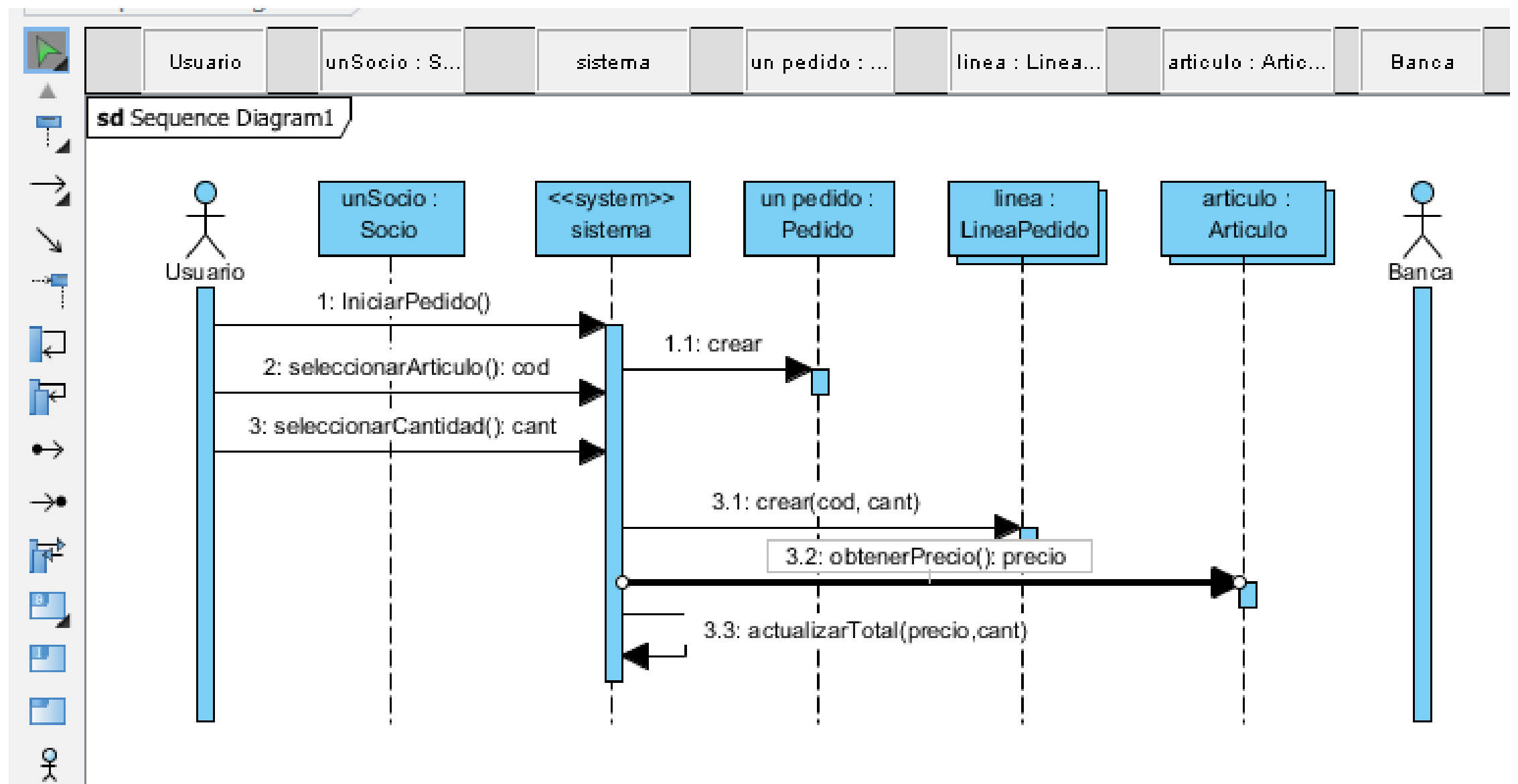
Description:

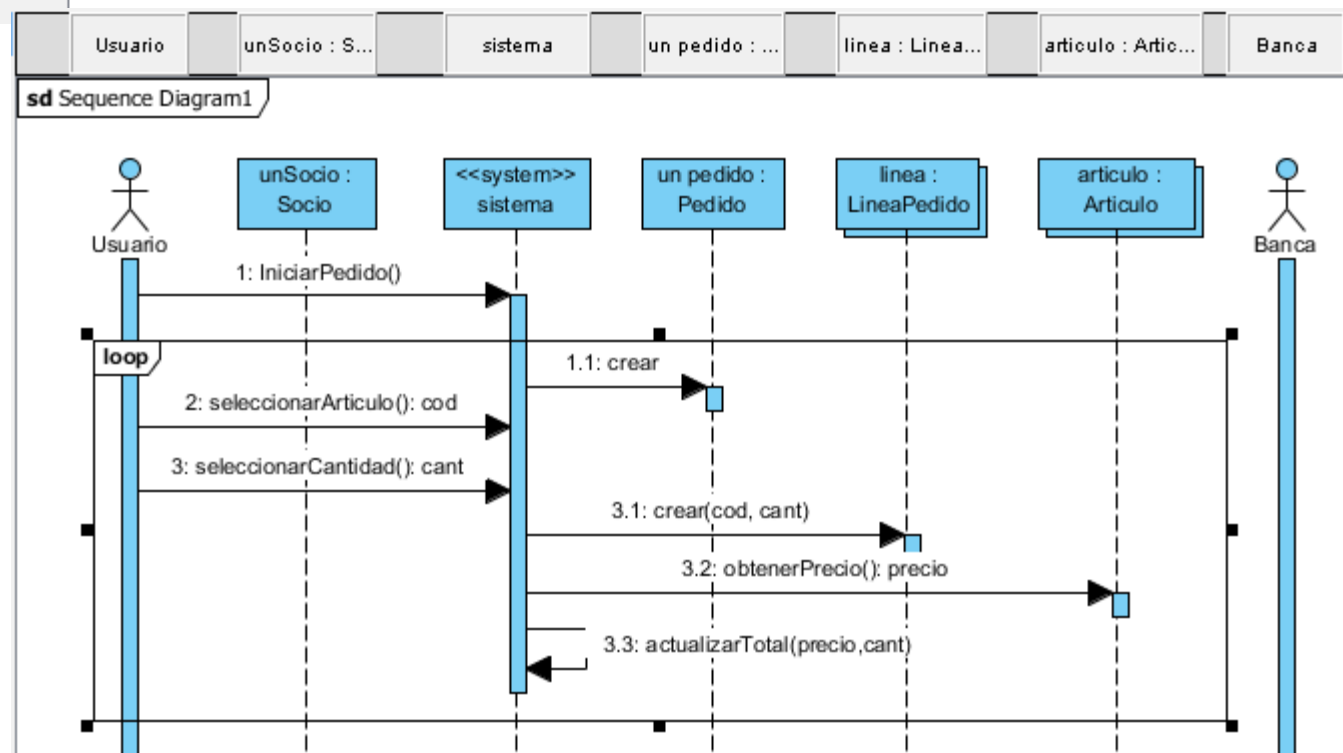
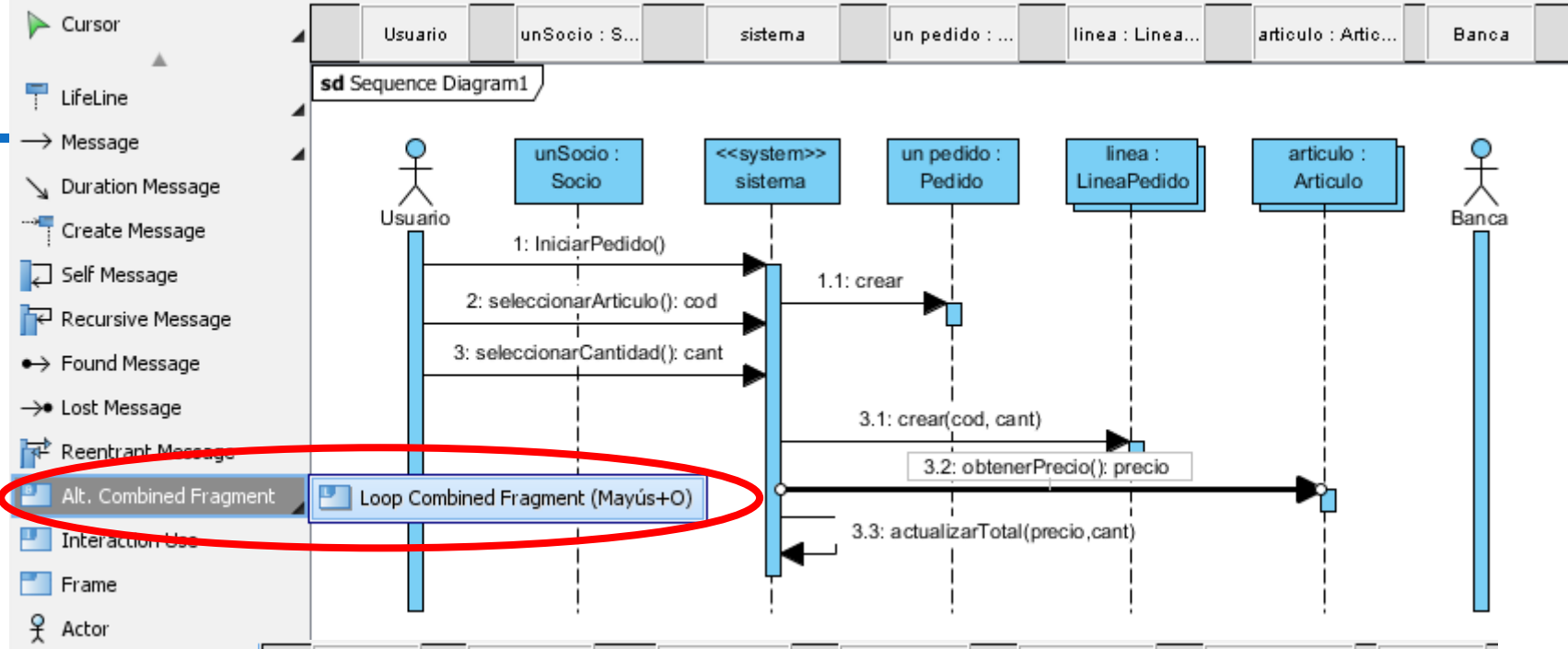
☐ Active ☐ Stereoped ☒ Multi-object

Reset OK Cancel Apply Help



Si necesito
devolver un
valor → Open
Specification>>
General>>
Return value...





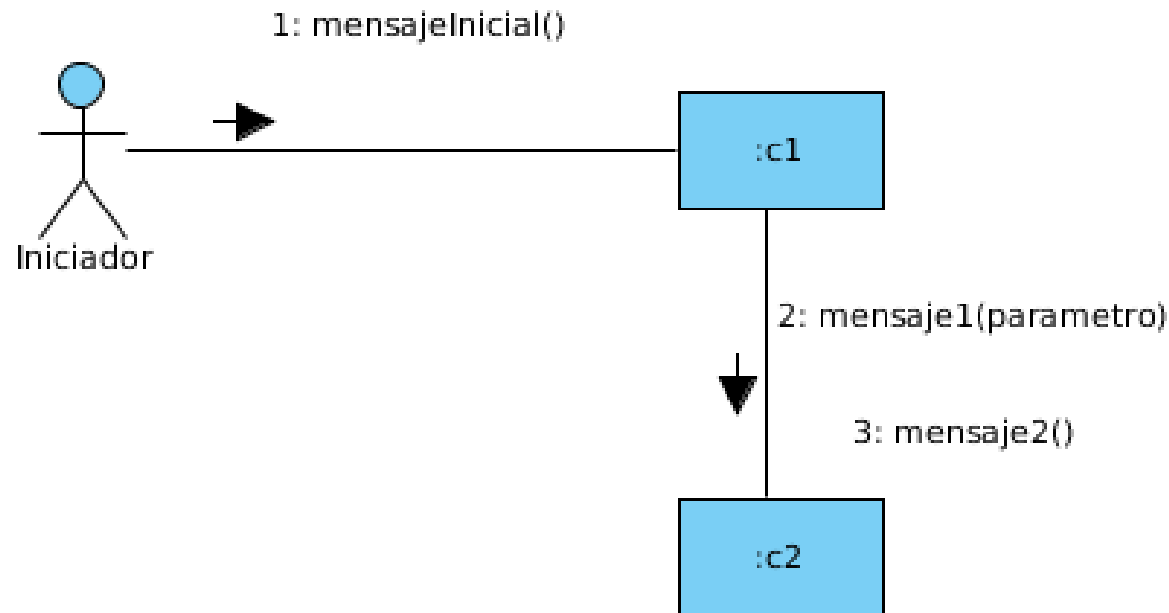
4.- Diagramas de colaboración

Diagramas de colaboración.

- **Objetivo.** Mostrar las interacciones entre los objetos del diagrama mediante el paso mensajes entre ellos.
- Grafo.
 - Nodos → objetos
 - Aristas → enlaces entre objetos - mensajes entre ellos. Los objetos se conectan mediante enlaces y se comunican a través de los mensajes.
- No se incluye una línea temporal
- Los mensajes son numerados para determinar su orden el tiempo.

Diagramas de colaboración

- **Actor Iniciador** manda un mensaje al objeto c1 que inicia el escenario, a continuación el objeto c1 envía el mensaje mensaje1 que lleva un parámetro al objeto c2 y después el mensaje mensaje2, que no tiene parámetros de nuevo al objeto c2.



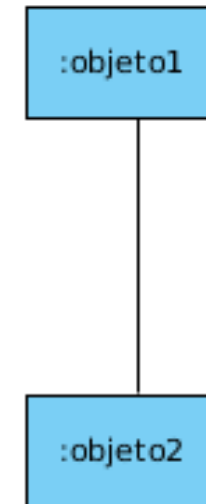
Diagramas de colaboración



- **Objeto** puede ser
 - cualquier instancia de las clases que hay definidas en el sistema, aunque también pueden incluirse objetos como
 - la interfaz del sistema,
 - o el propio sistema, si esto nos ayuda a modelar las operaciones que se van a llevar a cabo.
- Los objetos se representan mediante rectángulos en los que aparece uno de estos nombres.
 - **NombreClase:** directamente se puede utilizar el nombre de la clase a la que pertenece el objeto que participa en la interacción. Pero esta representación hace referencia a la clase, el resto son objetos.
 - **NombreObjeto:** se puede usar el nombre concreto del objeto que participa en la interacción, normalmente aparece subrayado.
 - **:nombreClase:** cuando se coloca el símbolo ":" delante del nombre de la clase quiere decir que hace referencia a un objeto genérico de esa clase.
 - **NombreObjeto:nombreClase:** hace referencia al objeto concreto que se nombre añadiendo la clase a la que pertenece.

Paso de mensajes

- Paso de mensajes → necesario que exista una asociación entre los objetos.
- En la imagen es posible el paso de mensajes entre el objeto objeto1 y objeto2, además de quedar garantizada la navegación y visibilidad entre ambos.

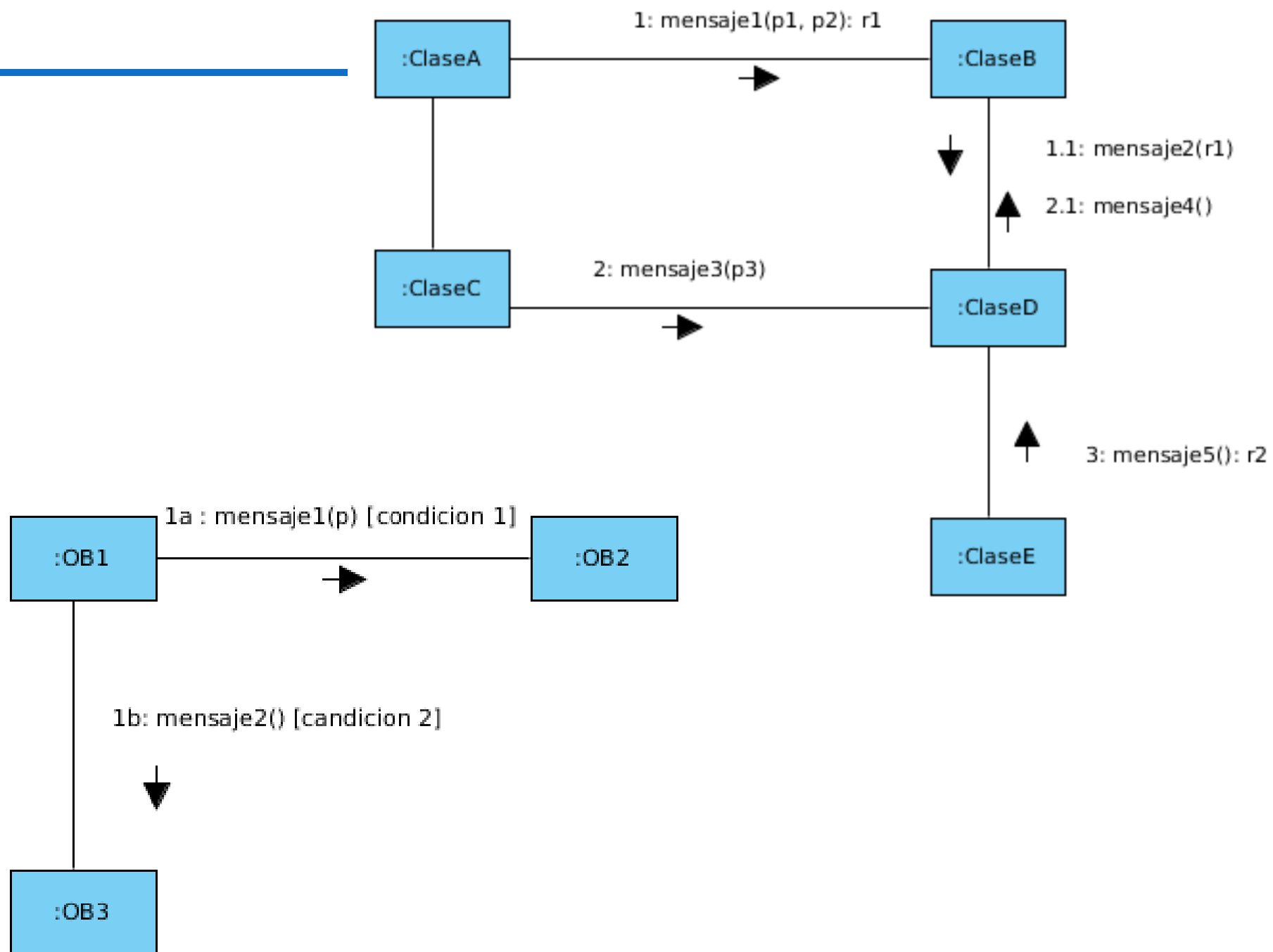


Paso de mensajes

- **Mensaje.** Especificación de una comunicación entre objetos que transmite información y desencadena una acción en el objeto destinatario.
- Sintaxis de un mensaje

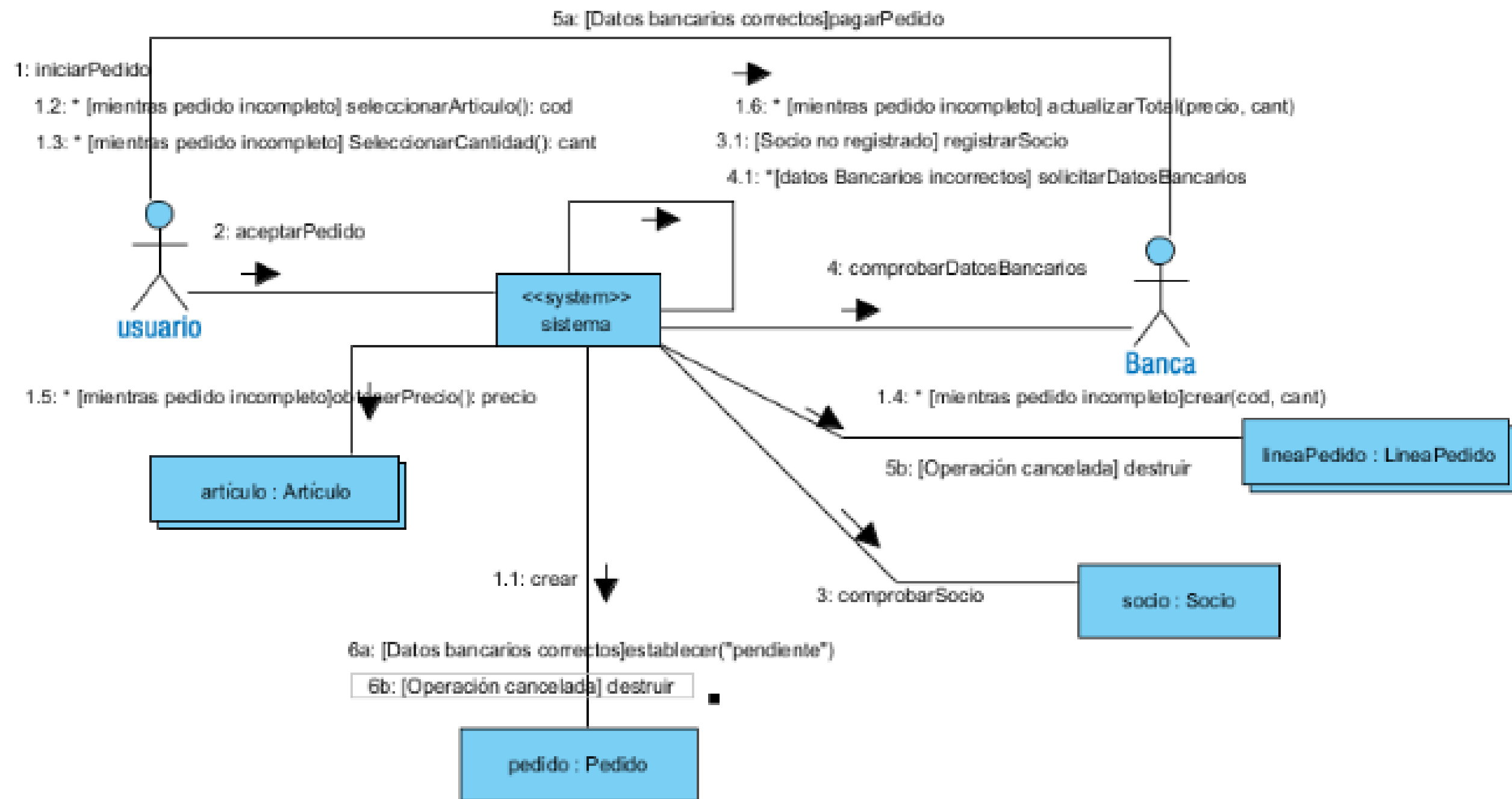
[secuencia][*] [Condición de guarda]{valorDevuelto} : mensaje (argumentos)

- **Secuencia:** representa el nivel de anidamiento del envío del mensaje dentro de la interacción. Los mensajes se numeran para indicar el orden en el que se envían, y si es necesario se puede indicar anidamiento incluyendo subrangos.
- *: indica que el mensaje es iterativo.
- **Condición de guarda:** debe cumplirse para que el mensaje pueda ser enviado.
- **ValorDevuelto:** lista de valores devueltos por el mensaje. Estos valores se pueden utilizar como parámetros de otros mensajes. Los corchetes indican que es opcional.
- **Mensaje:** nombre del mensaje.
- **Argumentos:** parámetros que se pasan al mensaje.
- Como se ve en el ejemplo se puede usar la misma asociación para enviar varios mensajes. Vemos que hay dos mensajes anidados, el 1.1 y el 2.1, se ha usado el nombre de los mensajes para indicar el orden real en el que se envían.



Elaboración de un diagrama de colaboración

- Las **actividades que se repiten** o pueden repetirse se marcan con un asterisco y su condición.
- Las **condiciones de guarda** se escriben en el mismo nombre del mensaje.
- El **flujo alternativo de eventos** según si el usuario cancela el pedido o no, obliga a modificar los números de secuencia de los mensajes 5 y 6, pasando a tener los mensajes 5a y 6a y 5b y 6b, según la condición. Puedes modificar el número de secuencia de los mensajes abriendo la especificación del diagrama, y seleccionando la pestaña Mensajes, donde puedes editar los números de secuencia haciendo doble clic sobre ellos.
- Al objeto "**sistema**" se le ha asignado el estereotipo system.



5.- Diagramas de actividad

Diagramas de actividad.

- Especialización del Diagrama de Estado
- Se compone de una serie de actividades y representa cómo se pasa de unas a otras.
- Las actividades se enlazan por **transiciones automáticas**.
- **Uso.** Modelar el flujo de control entre actividades concurrentes y secuenciales
- Define la lógica de control:
 - **En el modelado de los procesos del negocio.**
 - **En el análisis de un caso de uso.**
 - **En la comprensión del flujo de trabajo, a través de varios casos de uso.**
 - Aclaran cuando se trata de expresar **aplicaciones multihilos**.
- Grafo conexo
 - Nodos = Estados de actividad o de acción
 - Arcos = Transiciones entre estados.
 - Nodo inicial
 - Nodo final

Elementos del diagrama de actividad.

- **Estados.**

- **Estado de actividad:** Elemento **compuesto** cuyo flujo de control se compone de otros estados de actividad y de acción.
- **Estado de acción:** Estado que representa la ejecución de una acción atómica, que no se puede descomponer ni interrumpir, normalmente la invocación de una operación. Generalmente se considera que su ejecución conlleva un tiempo insignificante.
- Pueden definirse también otro tipo de estados:
 - **Inicial.**
 - **Final.**

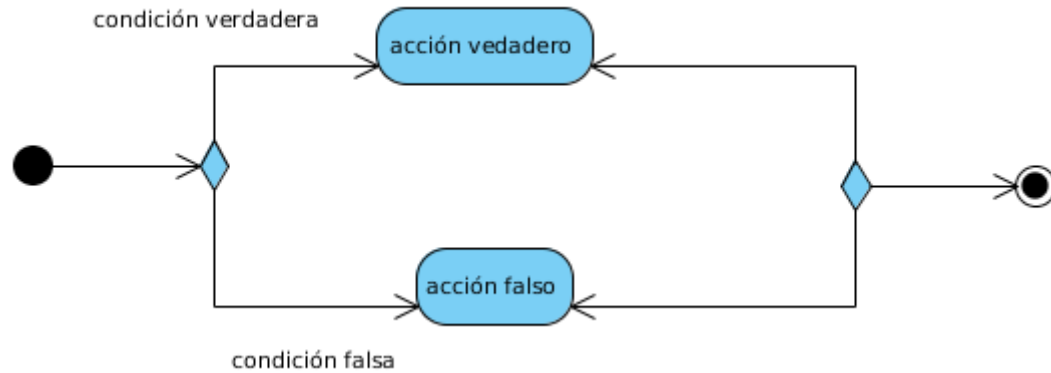


Elementos del diagrama de actividad.

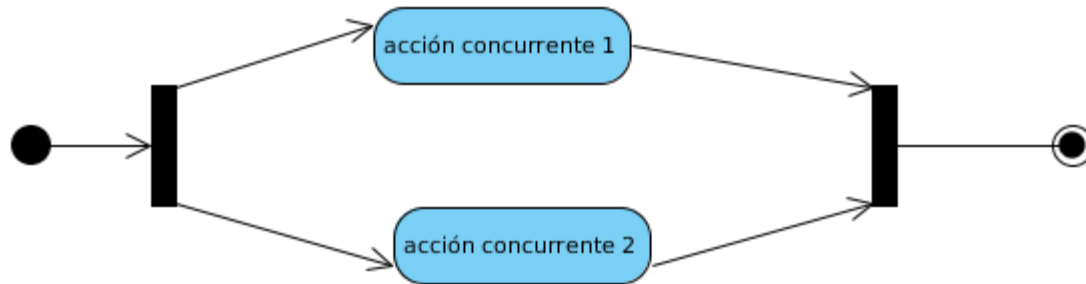
▪ **Transiciones**

- Paso de un estado a otro ante un evento en ciertas condiciones.
- Representación. Línea dirigida entre estados.
- Tipos de transacciones:
 - **Secuencial o sin disparadores.**
 - **Bifurcación (Decision node):** Especifica caminos alternativos
 - **Fusión (Merge node):** varios flujos de entrada en un único flujo de salida.
 - **División (Fork node):** sincronización o ejecución paralela de actividades.
 - **Unión (Join node):** flujos entrantes sincronizados, cada uno espera hasta que todos los flujos de entrada han alcanzado la unión.

Elementos del diagrama de actividad.



Bifurcación

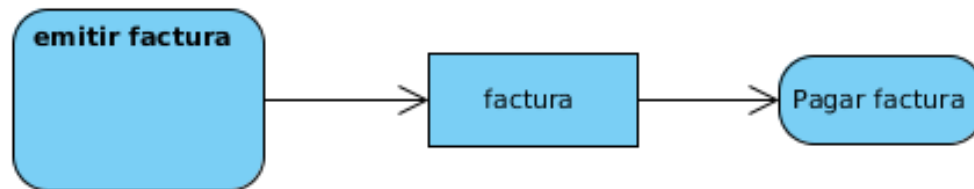


División

Elementos del diagrama de actividad

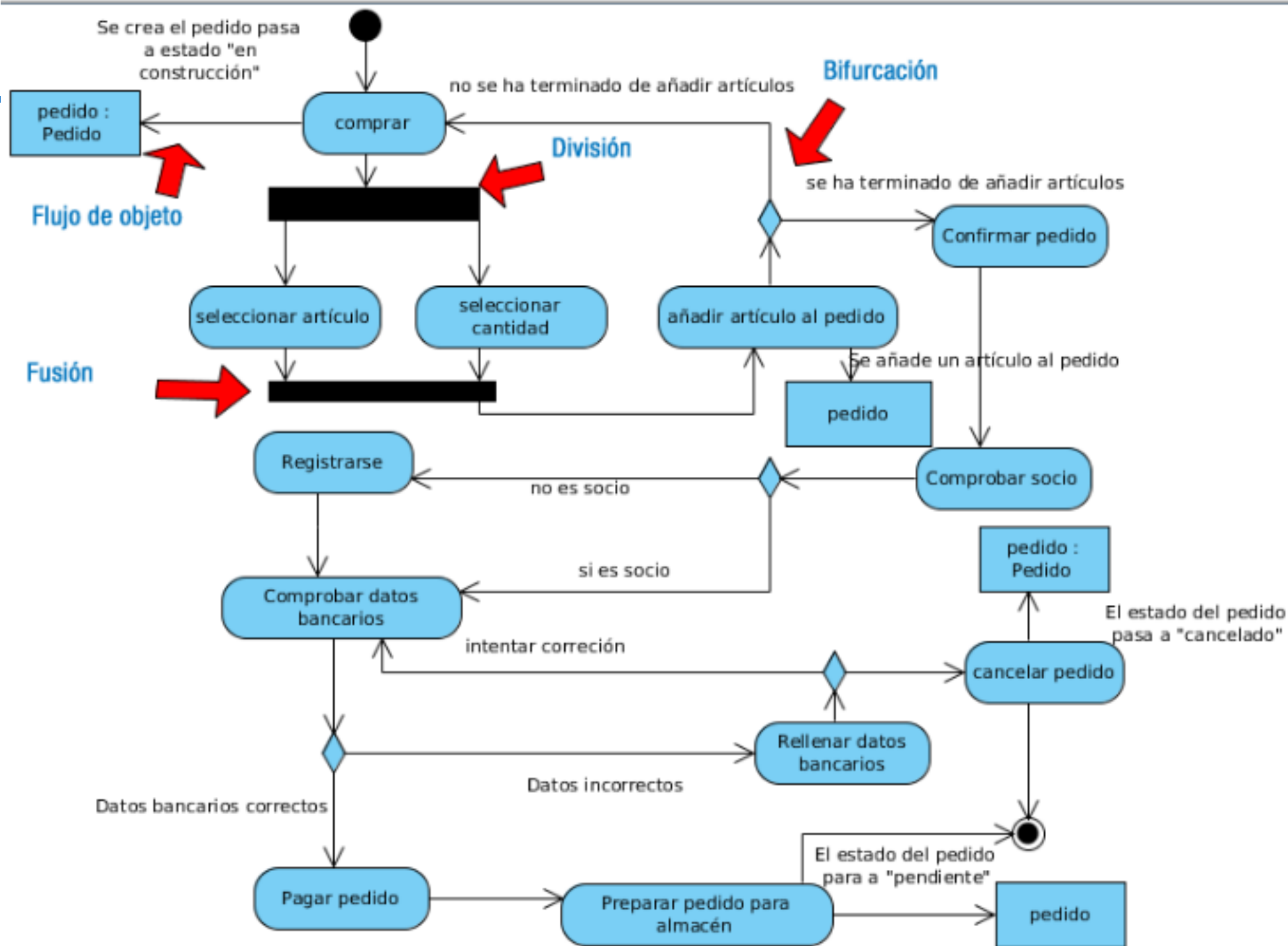
■ Objetos

- Manifestación concreta de una abstracción o instancia de una clase.
- Cuando interviene un objeto se utilizan flujos de objetos.
- Carriles o calles para ver **QUIENES** son los responsables de realizar las distintas actividades.
- Cada calle tiene un nombre único dentro del diagrama.
- Puede ser implementada por una o varias clases.
- Las actividades de cada calle se consideran independientes y se ejecutan concurrentemente a las de otras calles.



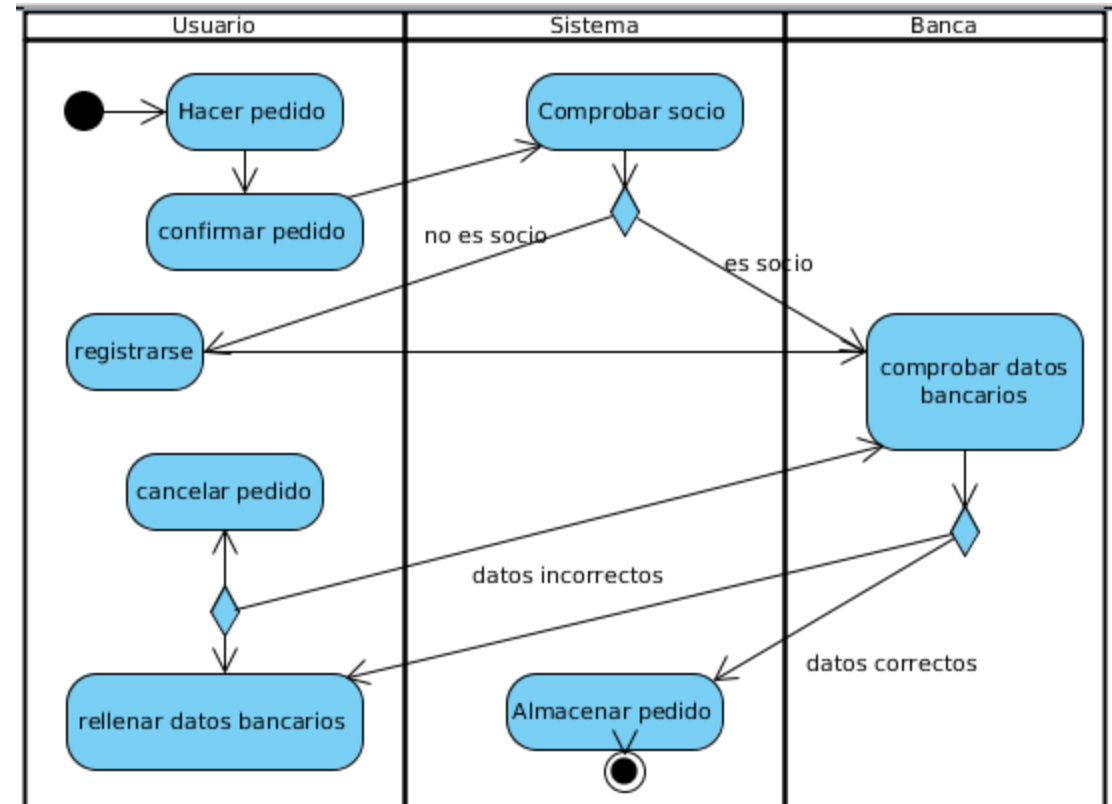
Elaboración de un diagrama de actividad

- En las bifurcaciones se ha añadido la condición que indica si se pasa a una acción o a otra.
- Las acciones *Seleccionar artículo* y *Seleccionar cantidad* se han considerado concurrentes.



Elaboración de un diagrama de actividad

- En este otro diagrama se simplifican las acciones a realizar y se eliminan los objetos para facilitar la inclusión de calles que indican quien realiza cada acción:
- Para añadir las calles en VP se utiliza la herramienta del panel "Vertical Swimlane"



6. Diagramas de estados

Diagramas de estados.

- Un **estado** de un objeto en un momento → valores de sus atributos.
- La **transición** de un estado a otro es momentánea y se produce cuando ocurre un determinado evento.
- Una **máquina de estados** procesa un **evento** cada vez y termina con todas las consecuencias del evento antes de procesar otro.
- Si ocurren dos eventos simultáneamente se procesan como si se hubieran producido en cualquier orden, sin pérdida de generalidad.
- Un **diagrama de máquina de estados** expresa el comportamiento de un objeto como una progresión a través de una serie de estados, provocada por eventos y las acciones relacionadas que pueden ocurrir.

Diagrama de estados

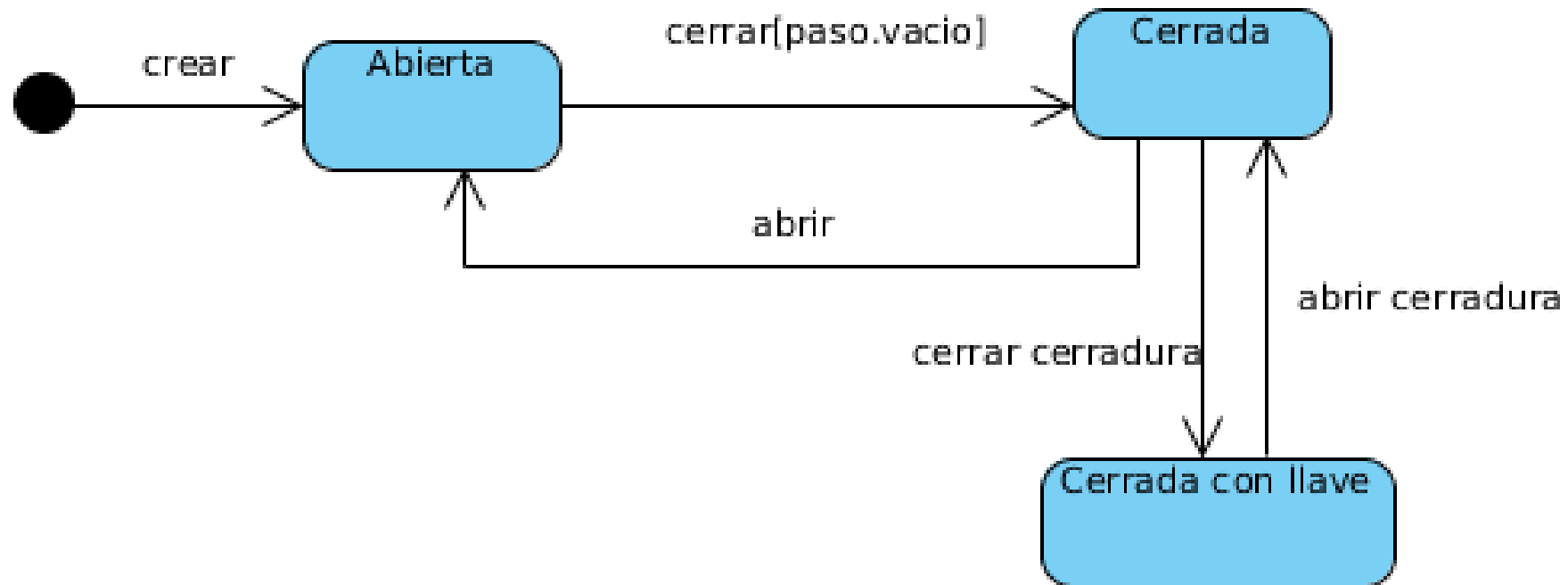


Diagrama de estados. Estados y eventos

- **Elementos** de un estado.
 - Nombre
 - Acciones entrada/salida.
 - Actividad a realizar.
 - Subestados, cuando el estado es complejo y necesita de un diagrama que lo defina.
 - Eventos diferidos.
- Existen dos tipos de estado especiales, estado inicial y estado final.
 - **Estado inicial:** el punto de partida por defecto para una transición cuyo destino es el límite de un estado compuesto.
 - **Estado final:** Estado especial dentro de un estado compuesto que, cuando está activo, indica que la ejecución del estado compuesto ha terminado y que una transición de finalización que sale del estado compuesto está activada.

Diagrama de estados. Estados y eventos

- Tipos de eventos: señales, llamadas, paso de tiempo y cambio de estado
 - **Señales (excepciones):** la recepción de una señal, que es una entidad a la que se ha dado nombre explícitamente (clase estereotipada), prevista para la comunicación explícita - y asíncrona- entre objetos. Es enviada por un objeto a otro objeto o conjunto de objetos. Las señales con nombre que puede recibir un objeto se modelan designándolas en un compartimento extra de la clase de ese objeto. Normalmente una señal es manejada por la máquina de estados del objeto receptor y puede disparar una transición en la máquina de estados.

Diagrama de estados. Estados y eventos

- **Llamadas:** la recepción de una petición para invocar una operación. Normalmente un evento de llamada es modelado como una operación del objeto receptor, manejado por un método del receptor y se implementa como una acción o transición de la máquina de estados.
- **Paso de tiempo:** representa el paso del tiempo (ocurrencia de un tiempo absoluto respecto de un reloj real o virtual o el paso de una cantidad de tiempo dada desde que un objeto entra en un estado). Palabra clave after: after (2 segundos); after 1 ms desde la salida de devInactivo.
- **Cambio de estado:** evento que representa un cambio en el estado o el cumplimiento de una condición. Palabra clave when, seguida de una expresión booleana, que puede ser de tiempo o de otra clase: when (hora = 11:30); when (altitud < 1000).

Diagrama de estados. Transiciones

- **Transición** de un estado A a un estado B, se produce cuando se origina el **evento** asociado y se satisface cierta **condición** especificada, en cuyo caso se ejecuta la acción de salida de A, la acción de entrada a B y la acción asociada a la transición.

Evento(argumentos) [condición] / accion

- Elementos de una transición: estados origen y destino, evento de disparo, condición de guarda, acción

Diagrama de estados. Transiciones

- **Estados origen y destino:** la transición se disparará si, estando en el estado origen se produce el evento de disparo y se cumple la condición de guarda (si la hay), pasando a ser activo el estado final.
- **Evento de disparo:** cuando se produce un evento, afecta a todas las transiciones que lo contienen en su etiqueta. Todas las apariciones de un evento en la misma máquina de estados deben tener la misma signatura. Los tipos de evento los hemos visto en el punto anterior.
- **Condición de guarda:** expresión booleana. Si es falsa, la transición no se dispara, y si no hay otra transición etiquetada con el mismo evento que pueda dispararse, éste se pierde.
- **Acción:** computación atómica ejecutable. Puede incluir llamadas a operaciones del objeto que incluye la máquina de estados (o sobre otros visibles), creación o destrucción de objetos o el envío de una señal a otro objeto.

Diagrama de estados. Transiciones

- Los diferentes estados en los que puede estar un pedido son:
 - **En creación:** es cuando se están seleccionando los productos que formará el pedido.
 - **Pendiente:** está en este estado desde que se confirma el pedido hasta que se selecciona para preparar su envío.
 - **En almacén:** está en este estado cuando es elaborado el paquete y se ha asignado a una ruta, hasta que se envía a través de la ruta que le corresponde.
 - **Servido:** Cuando el pedido es enviado. En este caso se envía una señal física desde el almacén cuando el transporte abandona el almacén.
 - **Cancelado:** puede llegarse a esta situación por dos motivos, o bien se cancela mientras se está haciendo por problemas con la tarjeta de crédito, o bien porque, una vez pendiente de su gestión el usuario decide cancelarlo, la diferencia fundamental entre ambos es que en el segundo caso hay que devolver el importe pagado por el pedido al socio que lo ha comprado.
 - Las transiciones entre estados se producen por llamadas a procedimientos en todos los casos, no intervienen cambios de estado o el tiempo, ni señales.

Diagrama de estados. Creación de un diagrama de estados

