



TEMA 4. DIAGRAMAS DE COMPORTAMIENTO. PARTE II.

CASO PRÁCTICO

María se ha dado cuenta de que los casos de uso permiten, de una manera sencilla, añadir información sobre qué hace el sistema, sin embargo, por completa que sea la descripción de la secuencia de eventos no permite incluir información útil, como los objetos que intervienen en las tareas, y como se comunican.

—Tendríamos que buscar la forma de representar como circula la información, que objetos participan en los casos de uso, qué mensajes envían, y en qué momento, esto nos ayudaría mucho a completar después el diagrama de clases.

Como siempre, la persona responsable del proyecto tiene una solución.

—Tendremos que investigar los diagramas de secuencia.

1. Diagramas de Secuencia

Los **diagramas de secuencia** se utilizan para **formalizar la descripción** de un **escenario** o conjunto de ellos **representando que mensajes fluyen en el sistema, así como quien los envía y quien los recibe**.

Los objetos y actores que forman parte del escenario se representan mediante rectángulos distribuidos horizontalmente en la zona superior del diagrama, a los que se asocia una **línea temporal vertical** (una por cada actor) de las que salen, en orden, los diferentes mensajes que se pasan entre ellos. Con esto el equipo de desarrollo puede hacerse una idea de las diferentes operaciones que deben ocurrir al ejecutarse una determinada tarea y el orden en que deben realizarse.

Los **diagramas de secuencia completan** a los diagramas de **casos de uso**, ya que permiten al equipo de desarrollo hacerse una idea de **qué objetos participan en el caso de uso y como interaccionan a lo largo del tiempo**.

1.1. Representación de objetos, línea de vida y paso de mensajes.

CASO PRÁCTICO

Ada, orienta a su equipo:

—Bien, ¿qué nos haría falta para poder representar la interacción de los objetos que participan en el caso de uso a lo largo del tiempo?

—Alguna manera de representar los objetos, el paso del tiempo y el paso de mensajes ¿no?

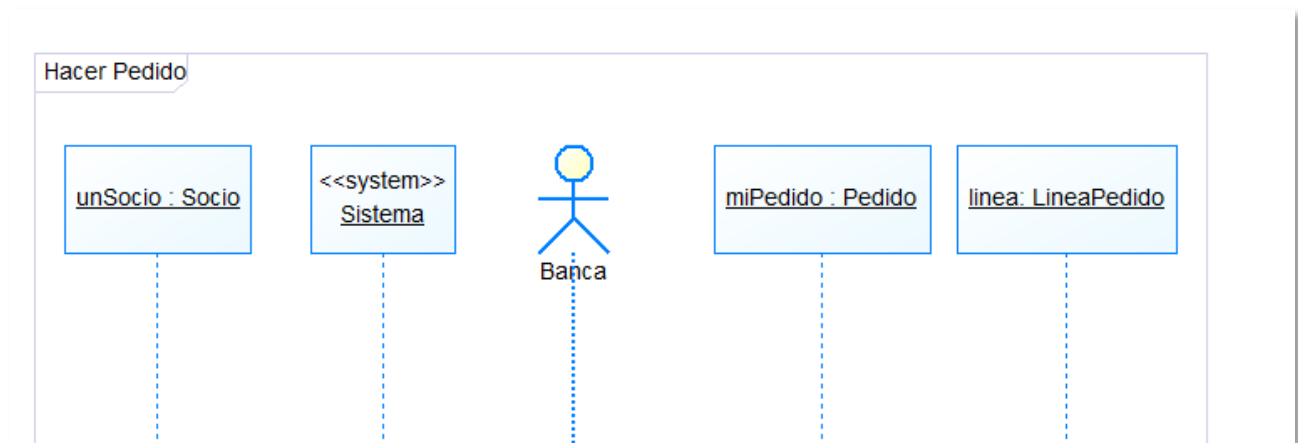
Representación de objetos y línea de vida.

En un **diagrama de secuencia se dibujan las entidades que participan dentro de rectángulos que se distribuyen horizontalmente**. De cada rectángulo o entidad sale una línea de puntos que **representa el paso del tiempo**, se les denomina **línea de vida** y representan que el objeto existe.

Para formar el nombre de una línea de vida de un objeto se usa el nombre del objeto, que es opcional, seguido del símbolo dos puntos y a continuación el nombre de la clase a la que pertenece.

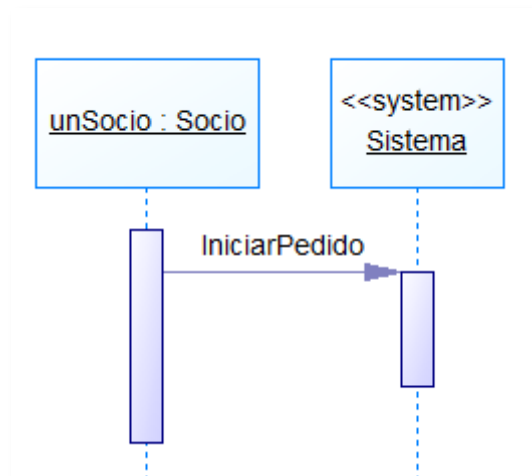
Una **línea de vida puede estar encabezada por otro tipo de instancias como el sistema o un actor** que aparecerán con su propio nombre. Usaremos el sistema para representar solicitudes al mismo, como por ejemplo pulsar un botón para abrir una ventana o una llamada a una subrutina.

Cuando tenemos un objeto que puede tener varias instancias, aparece como un rectángulo sobre otro, como en es el caso de las líneas del pedido que pueden ser varias.



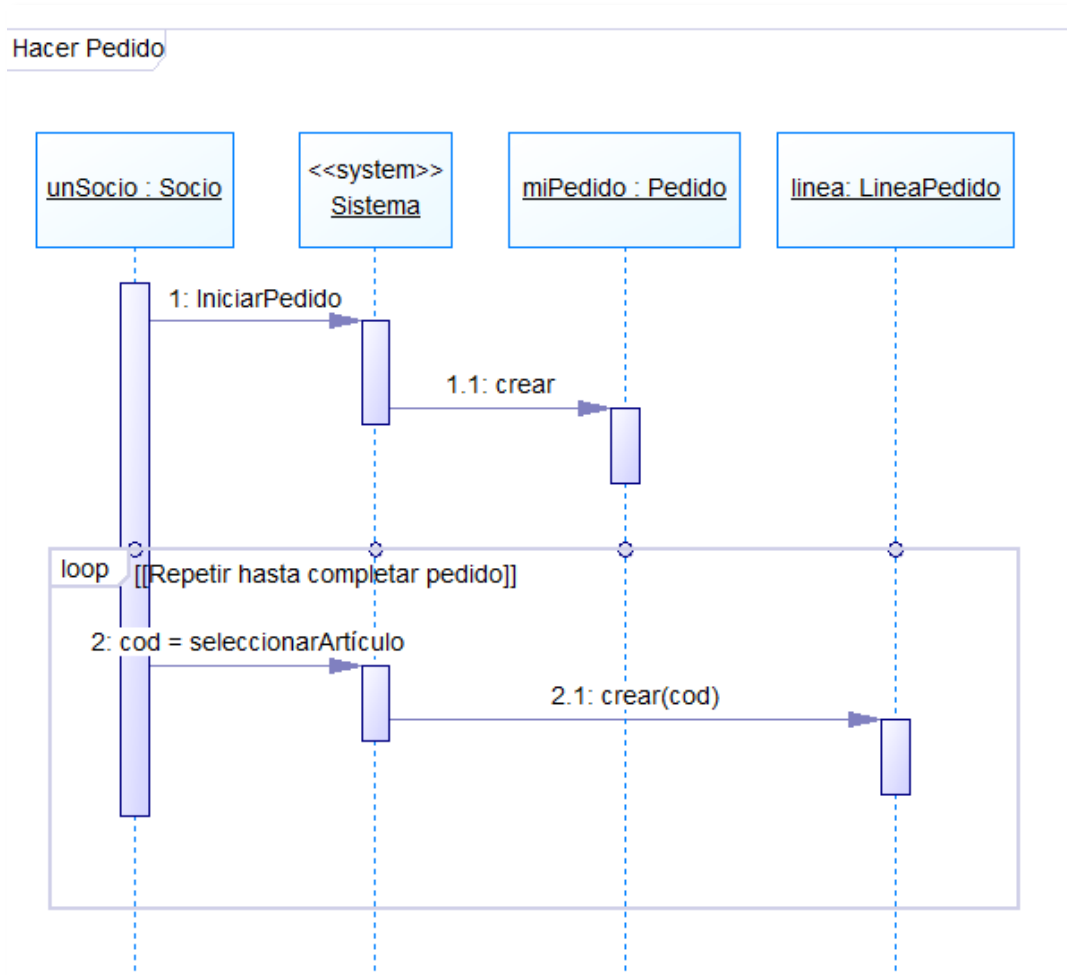
Invocación de métodos.

Los mensajes, que significan la invocación de métodos, se representan como flechas horizontales que van 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. Los mensajes se dibujan desde el objeto que envía el mensaje al que lo recibe, pudiendo ser ambos el mismo objeto y su orden viene determinado por su posición vertical, un mensaje que se dibuja debajo de otro indica que se envía después, por lo que no se hace necesario un número de secuencia.



Iteraciones y condicionales.

Además de presentar acciones sencillas que se ejecutan de manera secuencial también se pueden representar algunas situaciones más complejas como bucles usando marcos, normalmente se nombra el marco con el tipo de bucle a ejecutar y la condición de parada. También se pueden representar flujos de mensajes condicionales en función de un valor determinado.

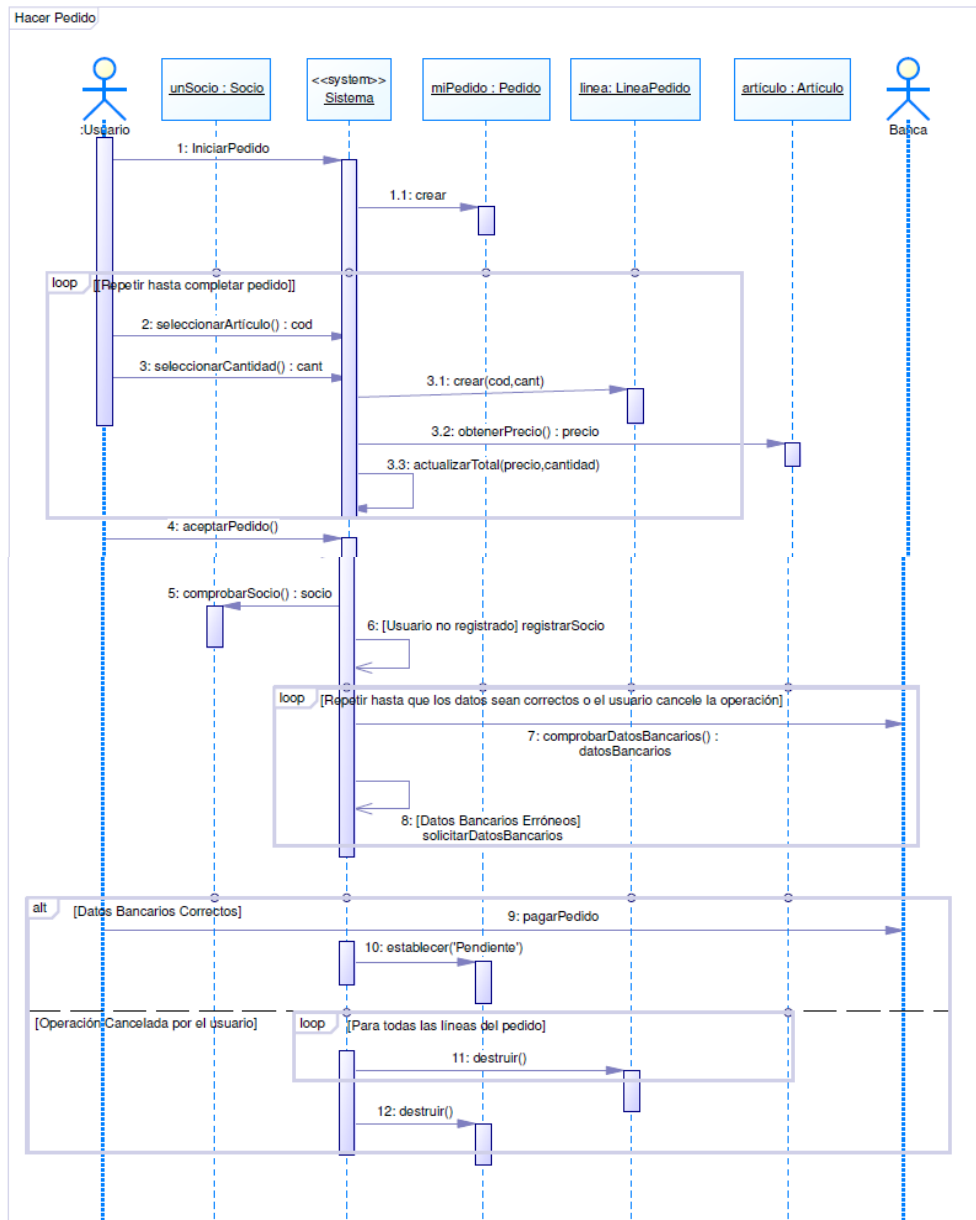


Por último, hay que destacar que se puede completar el diagrama añadiendo etiquetas y notas en el margen izquierdo que aclare la operación que se está realizando.

1.2. Elaboración de un diagrama de secuencia

Vamos a generar el diagrama de secuencia para el caso de uso **Hacer pedido**. En él se establece la secuencia de operaciones que se llevarán a cabo entre los diferentes objetos que intervienen en el caso de uso.

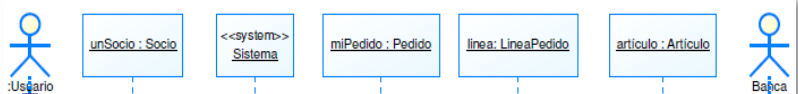
Este es el diagrama ya terminado, en él se han incluido todas las entidades (actores, objetos y sistema) que participan en el diagrama, y se han descrito todas las operaciones, incluidos los casos especiales, como es el registro de usuarios o la gestión de los datos bancarios. También incluye el modelado de acciones en **bucle**, como es la selección de artículos y de acciones regidas por condición, como es la posibilidad de cancelar el pedido si hay problemas con la tarjeta de crédito.



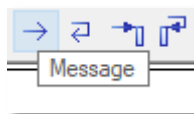
Pasos para la Creación del diagrama de Secuencia

Incluir entidades, actores, objetos y sistema. En el caso de *LineaPedido* y *Artículo*, se trata de un conjunto de objetos 'Multiobject' o 'Múltiple'.

Multiple: ☒



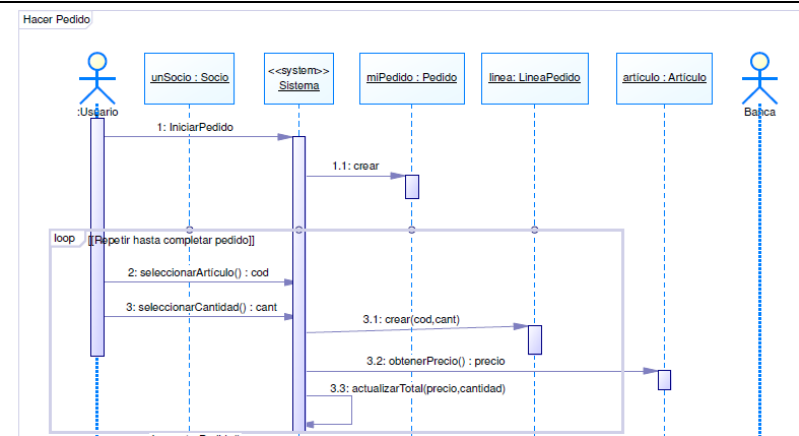
Mensajes



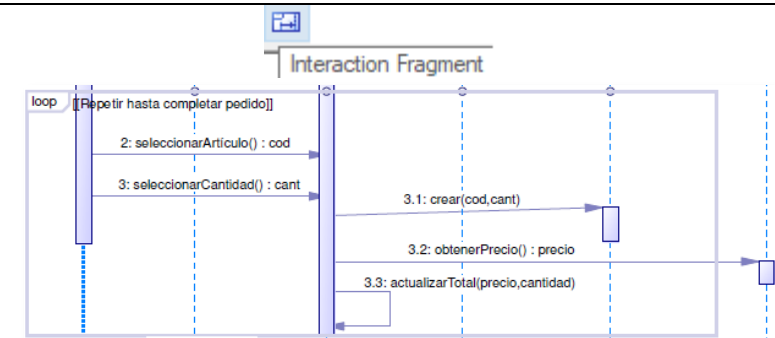
Los mensajes se añaden pinchando sobre el botón de mensaje, y seleccionando la flecha, entonces arrastraremos la flecha desde el origen a su fin. Ten en cuenta que un objeto puede enviarse un mensaje a si mismo seleccionando la opción *Self-message*.



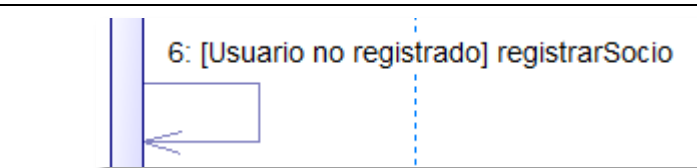
En esta imagen vemos los mensajes correspondientes al apartado de añadir los datos del pedido. Se selecciona el artículo y la cantidad, se crea una nueva línea de pedido y se recuperan los datos del artículo para obtener su precio y actualizar el total. Puesto que este proceso lo podemos repetir en más de una ocasión lo meteremos en un bucle (loop).



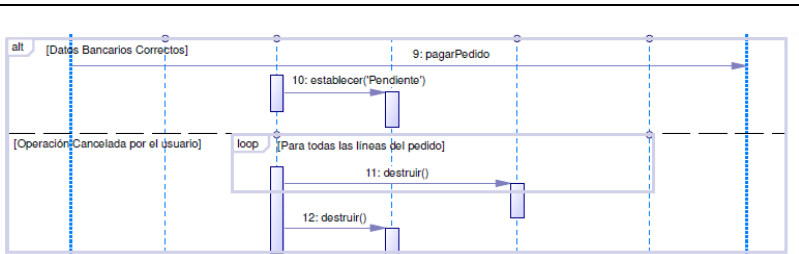
Para añadir un bucle seleccionamos la opción 'Interaction Fragment'.



Añadiremos condiciones de guarda cuando queramos indicar que un mensaje se enviará sólo si se cumple cierta condición, por ejemplo, sólo registraremos a un usuario si no es socio ya.



También se pueden incluir condiciones más elaboradas que implique una bifurcación en el flujo de eventos, como es el caso de la comprobación de la tarjeta, si todo marcha bien se finalizará la creación del pedido, si no, el usuario puede cancelar la operación sin guardar nada.



2. Diagramas de Colaboración

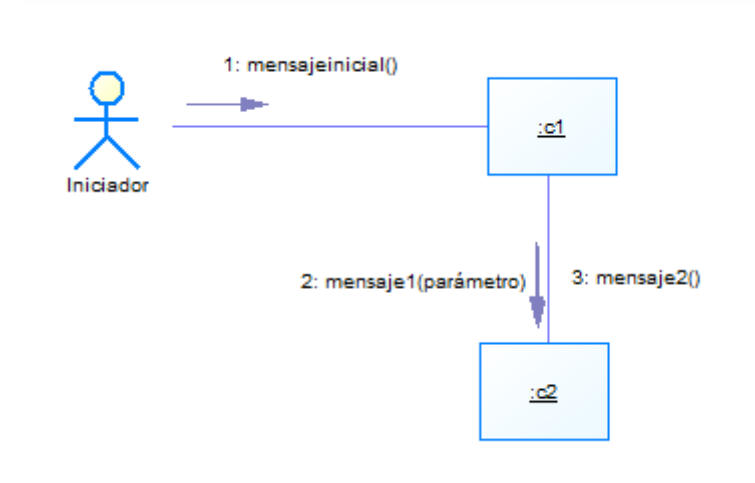
CASO PRÁCTICO.

El diagrama de secuencia ha aportado información muy valiosa sobre la circulación de mensajes en los casos de uso, sin embargo, estaría bien poder mostrar esta información de otra forma en la que se apreciase mejor el anidamiento de los mensajes, y el flujo de control entre objetos, ¿cierto?

Los **diagramas de colaboración** (También llamados **diagramas de Comunicación**) son un **complemento para los de secuencia cuyo objetivo es mostrar las interacciones entre los objetos** del diagrama mediante el **paso de mensajes** entre ellos. Las **interacciones entre los objetos** se describen en forma de **grafo** en el que los nodos son objetos y las aristas son enlaces entre objetos a través de los cuales se pueden enviar mensajes entre ellos. Los objetos se conectan mediante enlaces y se comunican a través de los mensajes.

A diferencia de los diagramas de secuencia, no se incluye una línea temporal; los mensajes son numerados para determinar su orden en el tiempo.

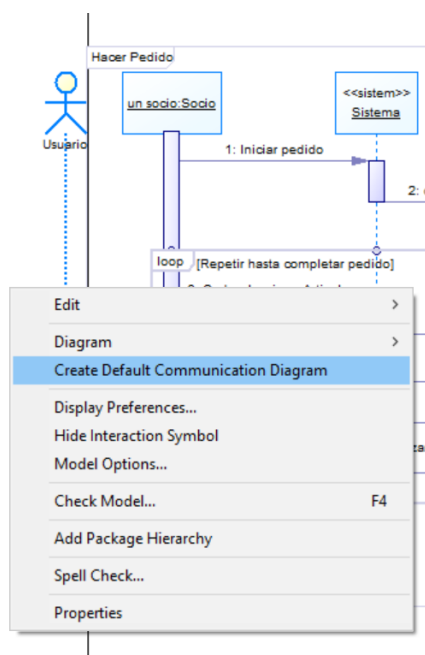
En este diagrama de colaboración el actor **Iniciador** manda un mensaje al objeto **c1** que inicia el escenario, a continuación, el objeto **c1** envía el mensaje **mensaje1** que lleva un parámetro al objeto **c2** y después el mensaje **mensaje2**, que no tiene parámetros de nuevo al objeto **c2**.



Los **diagramas de colaboración y secuencia** utilizan los mismos elementos, pero distribuyéndolos de forma diferente, ¿crees que son semejantes?

Ambos diagramas representan la misma información, representan entidades del sistema y los mensajes que circulan entre ellas, además **en ambos casos es fácil detectar la secuencialidad bien por el uso de líneas de vida, bien por los números de secuencia**. De hecho, en algunas aplicaciones para el desarrollo de estos diagramas es posible crear un diagrama a partir del otro. **Desde PowerDesigner en el mismo**

diagrama de secuencia pulsando el botón derecho y eligiendo “Create Default Communication Diagram” va a generar de forma automática el Diagrama de Colaboración.

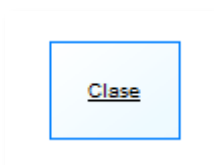


2.1. Representación de objetos.

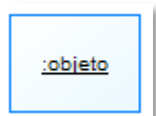
Un objeto puede ser cualquier instancia de las clases que hay definidas en el sistema, aunque también pueden incluirse objetos como la interfaz del sistema, o el propio sistema, si esto nos ayuda a modelar las operaciones que se van a llevar a cabo.

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.



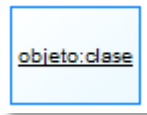
- **:nombreObjeto :** se puede usar el nombre concreto del objeto que participa en la interacción, normalmente aparece subrayado.



- **: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.



2.2. Paso de mensajes.

Para que sea posible el paso de mensajes es 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.



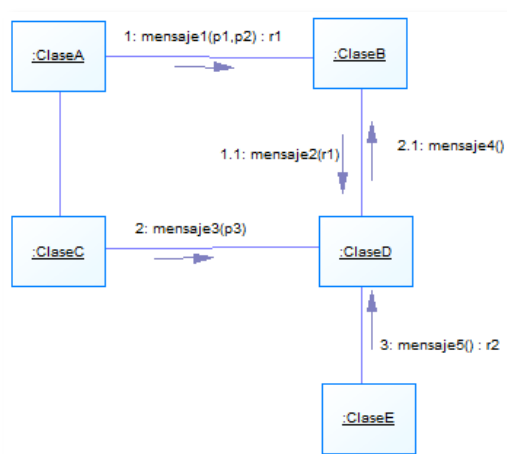
Un mensaje es la especificación de una comunicación entre objetos que transmite información y desencadena una acción en el objeto destinatario. La sintaxis de un mensaje es la siguiente:

<i>[secuencia][*] [Condición de guarda]{valorDevuelto} : mensaje (argumentos)</i>

donde:

- **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.

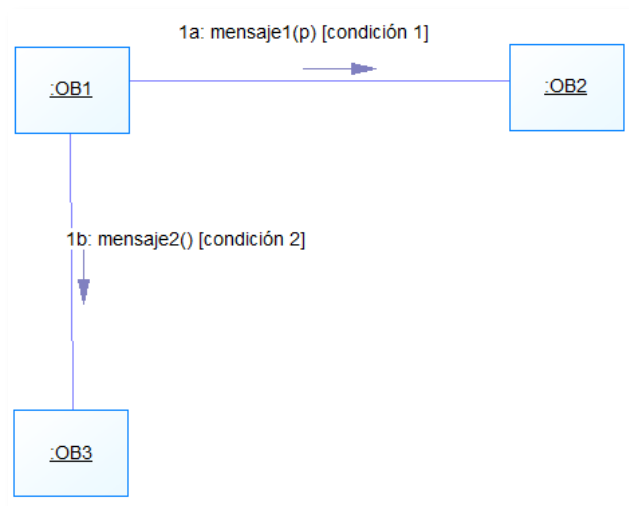
Como se ve en el ejemplo se puede usar la misma asociación para enviar varios mensajes. Vemos que hay dos mensajes anidados, el 1.1 y el 2.1, se ha usado el nombre de los mensajes para indicar el orden en el que se envían.



Los mensajes 1, 2 y 3 tienen parámetros, y los mensajes 1 y 5 devuelven un resultado.

Se contempla la bifurcación en la secuencia añadiendo una condición en la sintaxis del mensaje:

$[Secuencia][*][CondiciónGuarda]\{valorDevuelto\} : mensaje (argumentos)$

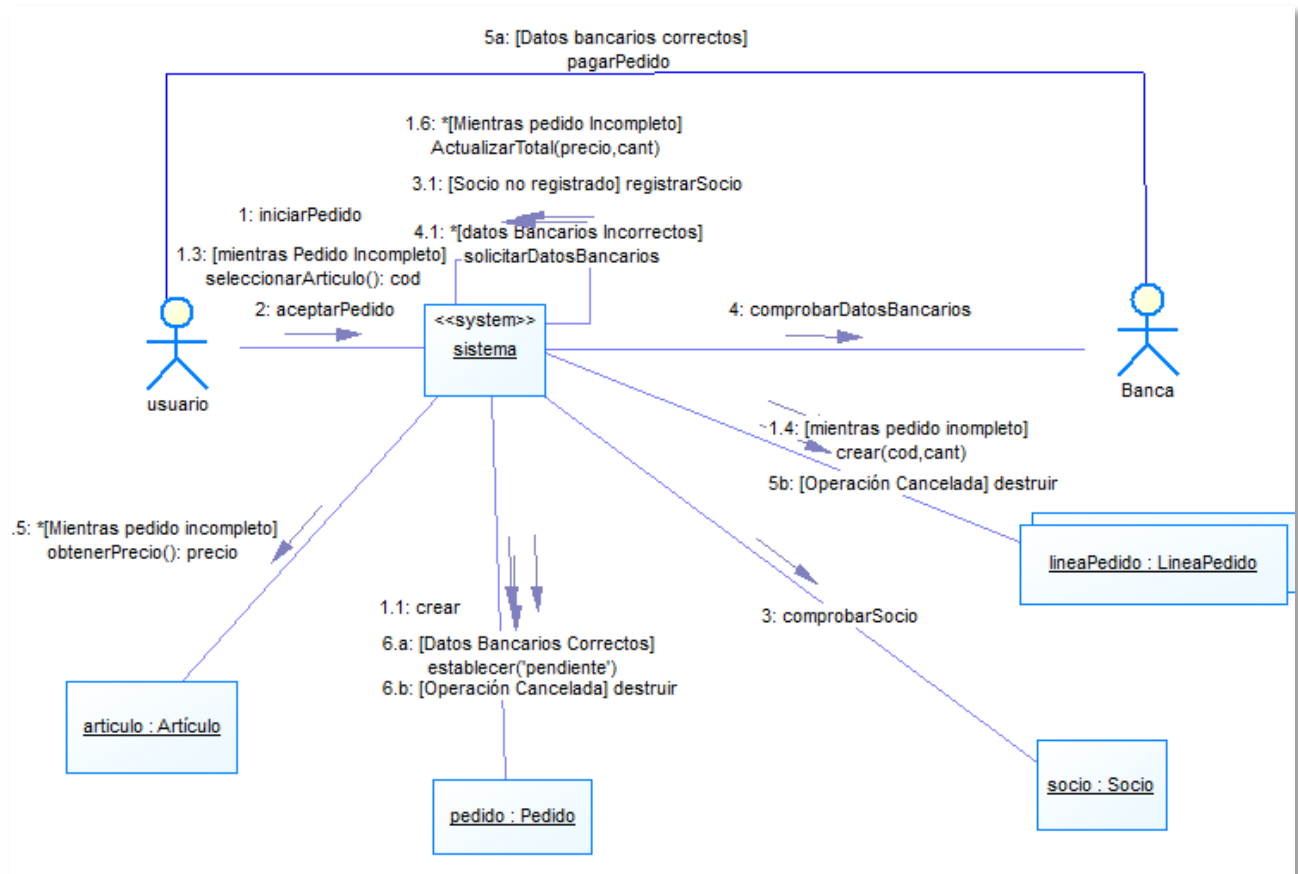


Cuando tenemos una **condición** se repite el número de secuencia y se añaden las condiciones necesarias, como vemos en la imagen según la condición se enviará el mensaje 1 o el 2, pero no ambos, por lo que coinciden en número de secuencia.

La iteración se representa mediante un * al lado del número de secuencia, pudiendo indicarse entre corchetes la condición de parada del bucle.

2.3. Elaboración de un diagrama de colaboración.

Este es el diagrama de colaboración del caso de uso Hacer pedido, se ha creado siguiendo el diagrama de secuencia, por lo que no te debe ser muy difícil seguirlo, de hecho, algunas aplicaciones para la creación de estos diagramas permiten la obtención de uno a partir de otro. Debes tener en cuenta que la aplicación modifica un poco la signatura de los mensajes, el valor devuelto se representa al final precedido de dos puntos.



Los aspectos más destacados son los siguientes:

- ✓ 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*.

3. Diagramas de actividad

CASO PRÁCTICO

¿Qué pasaría si quisiera representar sólo las acciones que tienen lugar, prescindiendo de quien las genera, solo el flujo de la actividad del sistema, qué pasa primero, qué ocurre después y qué cosas pueden hacerse al mismo tiempo?

Pasaría que tendrías que hacer un diagrama de actividad.

El **Diagrama de Actividad** es una especialización del **Diagrama de Estado**, organizado respecto de las acciones, que se compone de una serie de actividades y representa cómo se pasa de unas a otras. **Las actividades se enlazan por transiciones automáticas**, es decir, **cuando una actividad termina se desencadena el paso a la siguiente actividad**.

Se utilizan fundamentalmente para modelar el flujo de control entre actividades en el que se puede distinguir cuales ocurren secuencialmente a lo largo del tiempo y cuales se pueden llevar a cabo concurrentemente.

Un **diagrama de actividades** es un **grafo conexo en el que los nodos son estados**, que pueden ser de actividad o de acción y **los arcos son transiciones entre estados**. El grafo parte de un nodo inicial que representa el estado inicial y termina en un nodo final.

¿Por qué decimos que el diagrama de actividades visualiza el comportamiento desde otro punto de vista del resto de diagramas?

En los anteriores diagramas, tratábamos la interacción entre objetos y entidades, cual es el flujo de mensajes, en qué orden y en qué condiciones se envían, **en los diagramas de actividades se incluye el factor de la concurrencia, y trata sobre todo de expresar el flujo de las actividades, su elemento fundamental, y como se pasa de unas a otras**. Además, **también representa como se influye en los objetos**.

3.1. Elementos del diagrama de actividad.

Normalmente los diagramas de actividades contienen:

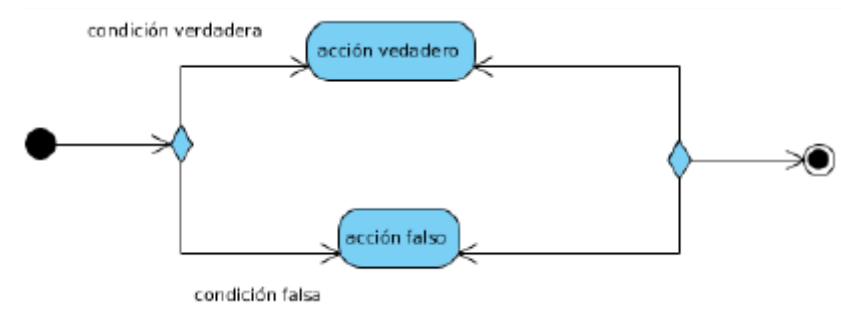
- ✓ **Estados** de actividad y estados de acción.
 - **Estado de actividad**: Elemento compuesto cuyo flujo de control se compone de otros estados de actividad y de acción.
 - **Estado de acción**: Estado que representa la ejecución de una acción atómica, que no se puede descomponer ni interrumpir, normalmente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

- **Final.**

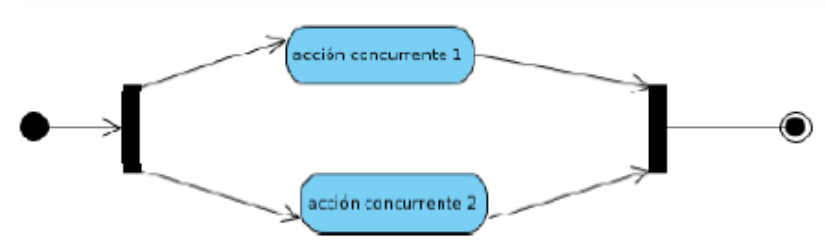


✓ **Transiciones:** Relación entre dos estados que indica que un objeto en el primer estado realizará ciertas acciones y pasará al segundo estado cuando ocurra un evento específico y satisfaga ciertas condiciones. Se representa mediante una línea dirigida del estado inicial al siguiente. Podemos encontrar diferentes tipos de transacciones:

- **Secuencial o sin disparadores:** Al completar la acción del estado origen se ejecuta la acción de salida y, sin ningún retraso, el control sigue por la transición y pasa al siguiente estado.
- **Bifurcación (Decision node):** Especifica caminos alternativos, elegidos según el valor de alguna expresión booleana. Las condiciones de salida no deben solaparse y deben cubrir todas las posibilidades (puede utilizarse la palabra clave else). Pueden utilizarse para lograr el efecto de las iteraciones.



- **Fusión (Merge node):** Redirigen **varios flujos de entrada en un único flujo de salida**. No tiene tiempo de espera ni sincronización.
- **División (Fork node):** Permiten expresar la **sincronización o ejecución paralela de actividades**. Las actividades invocadas después de una división son concurrentes.

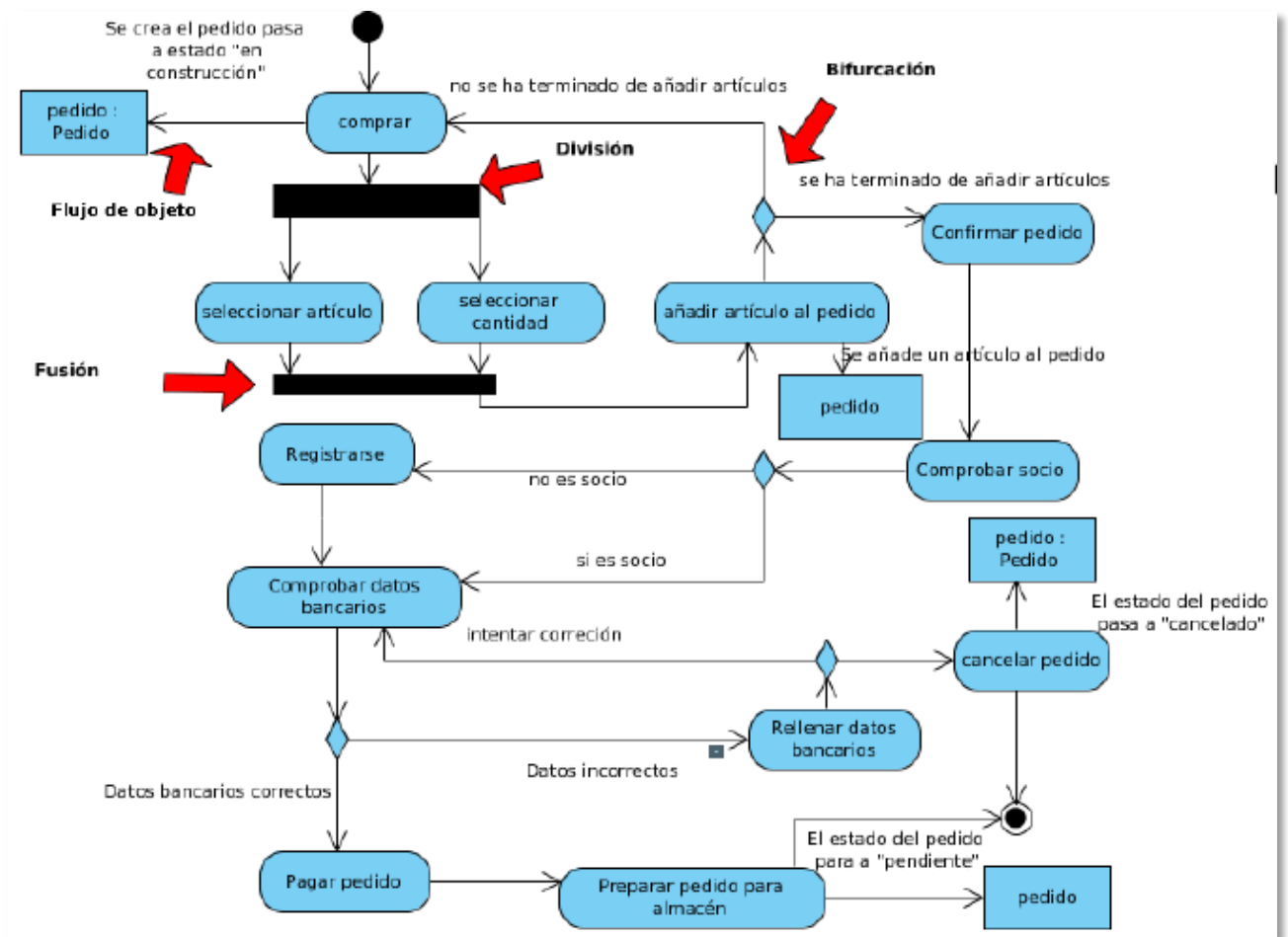


- **Unión (Join node):** Por definición, en la unión los flujos entrantes se sincronizan, es decir, **cada uno espera hasta que todos los flujos de entrada han alcanzado la unión**.

3.2. Elaboración de un diagrama de actividad.

El siguiente diagrama de actividad representa el caso de uso hacer pedido, en él aparecen los elementos que hemos visto en las secciones anteriores.

- ✓ 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.



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 incluir las calles dentro de *PowerDesigner*, haremos uso de 'Organization Unit Swimlane'.

