

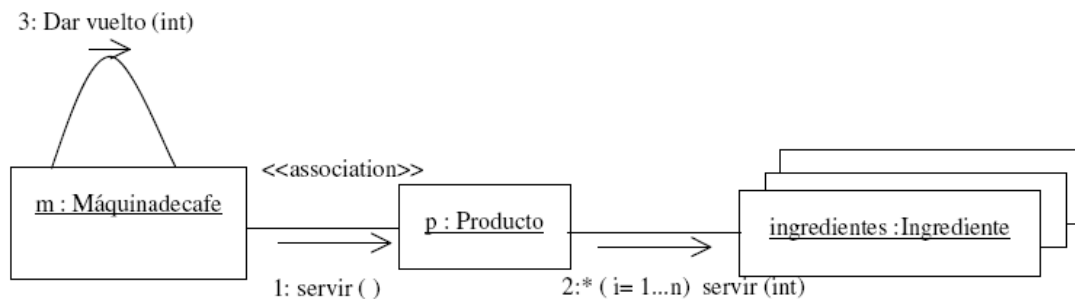
UNIDAD N° 7

Diagrama de Comunicación UML 2.0 (ex de colaboración)

Un diagrama de comunicación es una forma de representar interacción entre objetos, alterna al diagrama de secuencia.

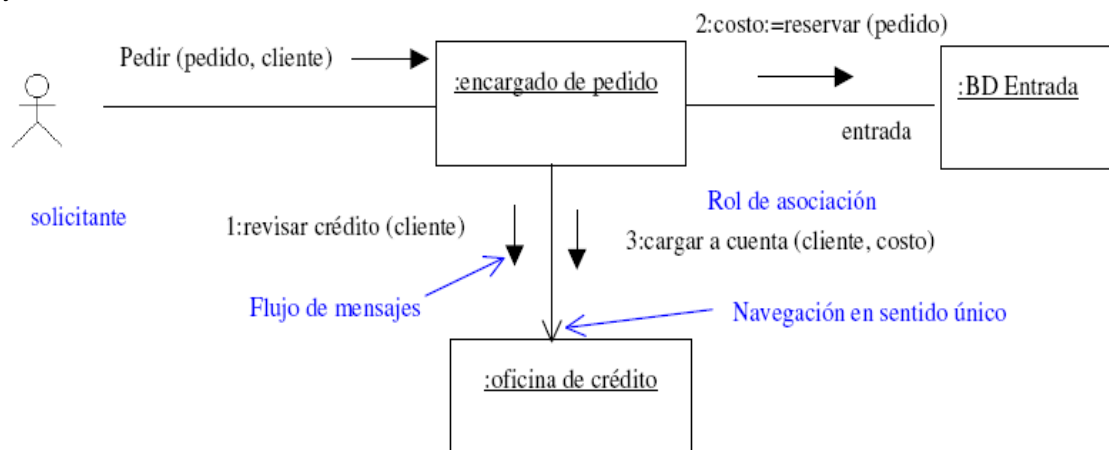
Es un diagrama de clases que contiene roles de clasificador y los roles de asociación en lugar de solo clasificadores y asociaciones. Cuando se instancia la colaboración, los objetos están ligados a los roles de clasificador y los enlaces están ligados a los roles de asociación. El rol de asociación también puede ser desempeñado por varios tipos de enlaces temporales, tales como argumentos de procedimientos o variables locales del procedimiento. Los símbolos del enlace pueden llevar estereotipos para indicar enlaces temporales (<<parameter>> o <<local>>) o llamadas al mismo objeto (<<self>>).

Pretende representar desde un punto de vista estático y dinámico los objetos implicados en la implementación de una función de aplicación. A diferencia de los diagramas de secuencia, pueden mostrar el contexto de la operación (cuáles objetos son atributos, cuáles temporales, etc.) y ciclos en la ejecución. Se toma como ejemplo el caso de uso PedirProducto ya descrito como diagrama de secuencia.



Solamente se representan los objetos que están implicados en la colaboración, aunque puede haber otros en el sistema completo.

Los diagramas de comunicación muestran interacciones entre objetos, insistiendo más particularmente en la estructura espacial estática que permite la colaboración de un grupo de objetos. Los diagramas de comunicación expresan a la vez el contexto de un grupo de objetos (a través de objetos y enlaces) y la interacción entre esos objetos (por la representación de envíos de mensajes). Los diagramas de comunicación son una extensión de los diagramas de objetos.



Es útil marcar los objetos en cuatro grupos:

- ✓ Los que existen con la interacción entera.
- ✓ Los creados durante la interacción (restricción {new})
- ✓ Los destruidos durante la interacción (restricción {destroyed} o {delete})
- ✓ Y los que se crean y destruyen durante la interacción (restricción {transient})

Durante el diseño, se puede empezar mostrando los objetos y los enlaces disponibles al comienzo de una operación y después decidir cómo el control puede fluir a los objetos correctos dentro del gráfico para implementar la operación.

1.1 Contexto

El contexto es una vista parcial estática de los objetos que se comunican para realizar una función. Corresponde pues, normalmente, a una parte, una vista local de un modelo de objetos que restringe o proporciona el ambiente para sus elementos. Los elementos estáticos representados (clases, instancias, atributos, enlaces, asociaciones, etc.), deben utilizarse obligatoriamente en la función descrita, lo cual permite mejorar notablemente la legibilidad del diagrama. El contexto de una interacción comprende los argumentos, las variables locales creadas durante la ejecución, así como los enlaces entre objetos que participan en la interacción.

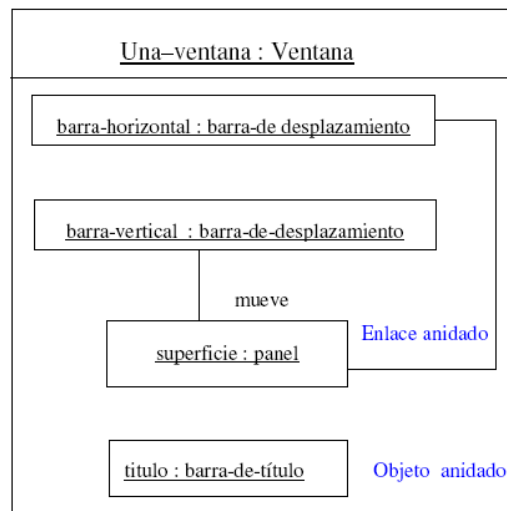
Aunque las comunicaciones muestran directamente la implementación de una operación, pueden también mostrar la realización de una clase entera. En este uso, muestran el contexto necesario para implementar todas las operaciones de una clase. Esto permite que el modelador vea los roles múltiples que los objetos pueden desempeñar en varias operaciones. Esta vista puede ser construida tomando la unión de todas las comunicaciones necesarias para describir todas las operaciones del objeto.

1.2 Objeto

Un objeto se representa con un rectángulo, que contiene el nombre y la clase del objeto en un formato nombreObjeto: nombreClase.

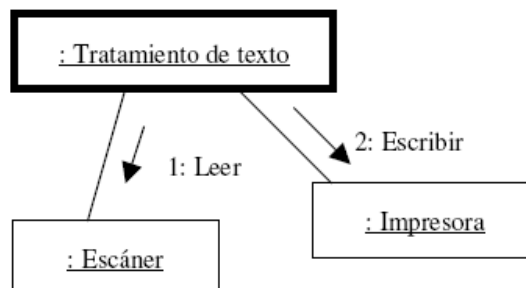
1.2.1 Objeto compuesto

En esta representación se muestran los objetos contenidos dentro del rectángulo que representa al objeto que los contiene. Representa un objeto de alto nivel construido con piezas unidas firmemente. Es una instancia de una clase compuesta, lo que implica la relación de agregación entre la clase y sus partes. Un ejemplo es el siguiente objeto ventana:



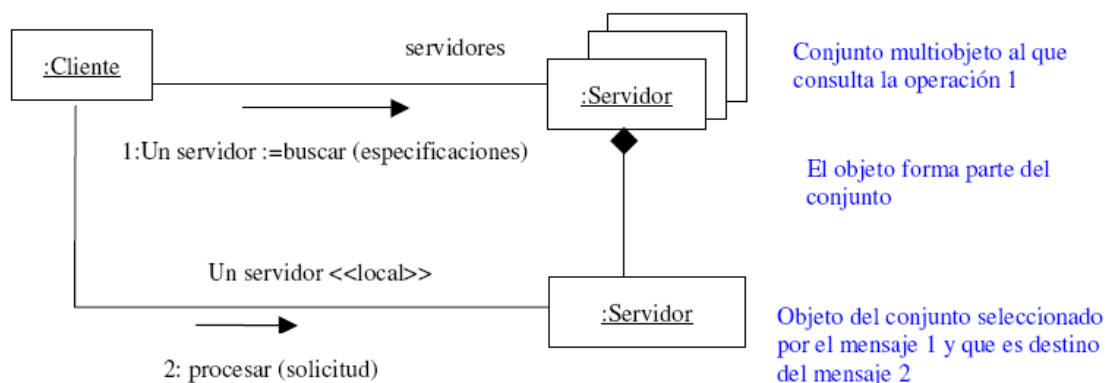
1.2.2 Objetos activos

Los objetos que poseen el flujo de control se llaman activos. Un objeto activo puede activar un objeto pasivo por el tiempo de una operación enviándole un mensaje. Una vez tratado el mensaje el flujo de control se restituye al objeto activo. En un entorno multitarea varios objetos pueden estar activos simultáneamente. Un objeto activo se representa por un rectángulo cuyo borde es más grueso que el de los objetos pasivos.



1.2.3. Objetos múltiples (Multiobjetos)

Alude a un rol de clasificador que denota un conjunto de objetos y no un objeto único. Generalmente es el conjunto de objetos del extremo muchos de una asociación.



1.3 Enlaces

Un enlace es una instancia de una asociación en un diagrama de clases. Se representa como una línea continua que une a dos objetos. Esta acompañada por un número que indica el orden dentro de la interacción y por un estereotipo que indica que tipo de objeto recibe el mensaje. Pueden darse varios niveles de subíndices para indicar anidamiento de operaciones. Los estereotipos indican si el objeto que recibe el mensaje es un atributo (association y se asume por defecto), un parámetro de un mensaje anterior, si es un objeto local o global.

La notación permite representar de manera condensada una familia de enlaces, instancias de una misma asociación. Esta aproximación es particularmente interesante cuando el grupo de objetos considerado se trata de manera uniforme siendo, por ejemplo, destino de un mismo mensaje.

1.4 Marcadores de creación y destrucción de objetos

Los diagramas de comunicación muestran simultáneamente las interacciones entre objetos y las relaciones estructurales que permiten estas interacciones. Los objetos y los enlaces creados o destruidos en el transcurso de una interacción pueden, respectivamente, contener las restricciones {nuevo} en inglés {new} o {destruido} en inglés {destroyed} o {delete}. Los objetos creados y luego destruidos dentro de la misma interacción, se identifican por la restricción {transitorio} en inglés {transient}.

1.5 Interacciones

Las interacciones ente objetos descritas en un diagrama de comunicación son secuencias de mensajes intercambiados por los objetos en el marco de la realización de una operación. Primero es necesario implementar el contexto, antes de definir las interacciones.

1.5.1 Mensajes

Los mensajes se muestran como flechas etiquetadas unidas a los enlaces. Cada mensaje tiene un número de secuencia, una lista opcional de mensajes precedentes, una condición opcional de guarda, nombre y lista de argumentos, y un nombre de valor de retorno opcional.

El número de serie incluye el nombre (opcional) de un hilo. Todos los mensajes del mismo hilo se ordenan secuencialmente. Los mensajes de diversos hilos son concurrentes a menos que haya una dependencia secuencial explícita. Pueden añadirse varios detalles de implementación, como por ejemplo una distinción entre mensajes síncronos y asíncronos.

1.5.1.1 Etiqueta del mensaje.

Las etiquetas tienen la siguiente sintaxis:

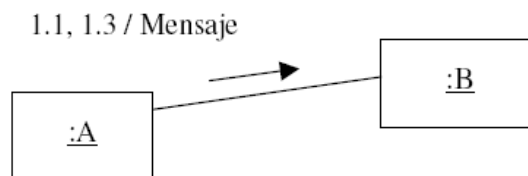
**Predecesor_{opt} condición-de-guarda_{opt} expresión-secuencia_{opt} lista-valores-
retorno:_{opt} nombre-mensaje (argumentos lista)**

- ✓ **Predecesor** (también llamada sincronización): en una colaboración, el predecesor es una lista de números de secuencia separados por comas y seguidos por una barra (/)
Número de secuencia _{lista} /

La cláusula se omite si la lista está vacía. Todo número de secuencia es *una expresión de secuencia* sin términos de recurrencia. Tiene que coincidir con el *número de secuencia* de algún otro mensaje.

- ✓ El significado es que el flujo de mensajes no queda habilitado mientras no se hayan producido todos los flujos de mensajes cuyos números de secuencia estén enumerados (un hilo puede ir más allá del flujo de mensajes y la condición de guarda seguirá cumpliéndose). Por lo tanto, **condición de guarda** representa una sincronización de hilos.

En el ejemplo siguiente, el mensaje se envía cuando los envíos 1.1 y 1.3 han sido satisfechos.



- ✓ **Expresión de secuencia**: la expresión de secuencia es una lista de términos de secuencia separados por puntos y seguida por dos puntos (':'). Cada término representa un nivel de anidamiento procedural dentro de la interacción global. Si es concurrente todo el control, entonces no se produce el anidamiento. Todos los términos de la secuencia poseen la siguiente sintaxis:

Etiqueta recurrencia _{opt}

En donde **etiqueta** es **entero** o bien **nombre**.

- El **entero** representa el orden secuencial del mensaje dentro del nivel inmediatamente superior de llamadas a procedimientos

Por ejemplo: el mensaje 3.1.4 sigue al 3.1.3 dentro de la activación 3.1.

- El **nombre** representa un hilo de control concurrente. Los mensajes que difieren en el nombre final son concurrentes en ese nivel de anidamiento.

Por ejemplo: el mensaje 3.1a y el mensaje 3.1b son concurrentes dentro de la activación 3.1. todos los hilos de control son iguales dentro de cada profundidad de anidamiento.

- La **recurrencia** representa la ejecución condicional o iterativa. Esto indica cero o más mensajes que se ejecutan, dependiendo de las condiciones.

Las opciones son:

*[**cláusula de iteración**] → una iteración

Una iteración representa una secuencia de mensajes en la profundidad de anidamiento dada. La **cláusula de iteración** se puede omitir (en cuyo caso las condiciones de iteración no estarán especificadas). La cláusula de iteración debería expresarse en pseudocódigo o en algún lenguaje de programación real; UML no prescribe su formato. Un ejemplo podría ser :

***[i:=1...n]**

La notación de iteración supone que los mensajes de la iteración se ejecutan secuencialmente. También existe la posibilidad de ejecutarlos concurrentemente. La notación empleada consiste poner una doble línea vertical, indicadora de paralelismo, después del asterisco: (* ||).

Una condición representa un mensaje cuya ejecución es contingente respecto a la veracidad de la cláusula de condición. La **cláusula de condición** debería expresarse en pseudocódigo o en un lenguaje de programación real; UML no prescribe su formato. Un ejemplo podría ser:

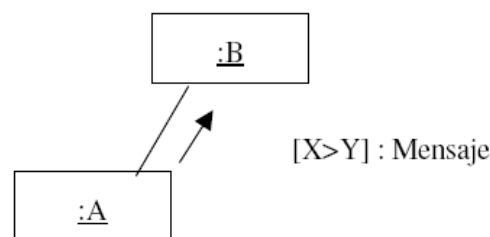
[x > y]

Obsérvese que las bifurcaciones se anotan igual que una interacción sin asterisco. Puede pensarse que se trata de una iteración limitada a un solo caso.

La cláusula de condición valida o no el envío de mensajes contenidos por el bloque.

La cláusula de condición se expresa en un formato libre:

- **Signatura:** una signatura es una cadena que indica el nombre, argumentos y valor proporcionado por una operación, mensaje o señal. Posee las siguientes propiedades:
- **Lista-de-valores-proporcionados:** lista de nombres, separados por comas que denota los valores proporcionados por el mensaje en la ejecución subsiguiente de la interacción global. Si el mensaje no proporciona ningún valor se omite.
- **Lista-valores-retorno (resultado):** El resultado está constituido por la lista de valores devueltos por el mensaje. Estos valores pueden utilizarse como parámetros de los otros mensajes comprendidos en la interacción. Este campo no existe en ausencia de valores devueltos. El formato de este campo es libre:



- **Nombre-mensaje:** es el nombre del evento surgido en el objeto destino (suele ser el evento que solicita la realización de una operación) . Puede implementarse de varias maneras: una de las cuales es una llamada a una operación. Si se implementa mediante una llamada a procedimiento, entonces se trata del nombre de la operación y esa operación tiene que estar definida en la clase receptor, o debe ser heredada por él. En otros casos puede ser el nombre de un evento que surge en el objeto receptor. En la práctica habitual, con sobrecarga de procedimientos, para identificar a una operación se necesita tanto el nombre del mensaje como la lista de tipos de los argumentos.
- **Lista-argumentos:** Se trata de una lista de parámetros del mensaje; es una lista de argumentos, separados por comas, encerrados entre paréntesis. Se pueden utilizar los parámetros aún cuando la lista esté vacía. Los argumentos y el nombre del mensaje identifican de manera única la acción que debe emprenderse en el objeto destinatario. Los argumentos pueden contener valores devueltos por mensajes enviados anteriormente, así como expresiones de navegación construidas a partir del objeto fuente.

Notación:

Sincronización: /
 Secuencia: .
 Iteración: *[]
 Iteración en paralelo: * ||
 Resultados: :=
 Argumentos: ()

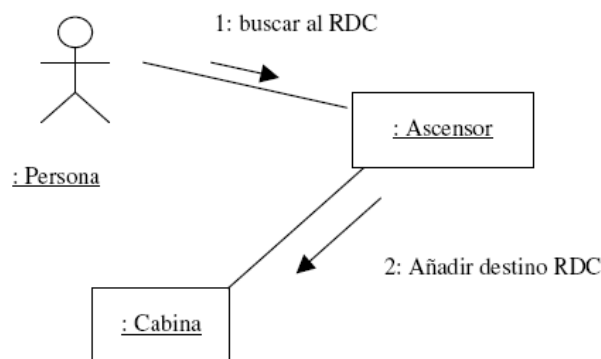
Ejemplos de mensajes

Las expresiones siguientes dan algunos ejemplos de la sintaxis de envío de mensajes:

- **mensaje simple:** 4 : Visualizar (x, y)
- **mensaje anidado:** 3.3.1 : Visualizar (x,y)
- **mensaje anidado con valor devuelto:**
 - 4.2 : edad : = sustraer (hoy, fechaDeNacimiento)
- **mensaje condicional:** [Edad >= 18 años] 6.2 Votar ()
- **sincronización con otros flujos de ejecución:** 4.a, b.6 / c.1 : Encender (lámpara)
- **iteración:** 1 * : Lavar ()
- **iteración paralela :** 3.a , 3.b / 4 * || [i : = 1..n]: Apagar ()

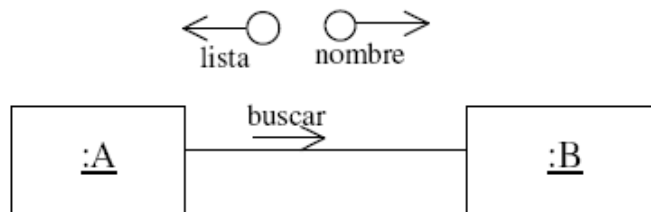
1.5 Actor

La notación permite incluir a un actor en un diagrama de comunicación para representar el desencadenamiento de las interacciones por un elemento externo al sistema. Gracias a este artificio la interacción puede describirse de manera más abstracta, sin entrar en detalles de los objetos de la interfaz del usuario. El primer mensaje de la interacción es enviado por el actor, representado bien por el símbolo gráfico de los actores del modelo de los casos de uso, o bien por un objeto con un estereotipo que precisa su calidad de actor. El siguiente diagrama muestra un fragmento de interacción, que corresponde a una llamada de cabina de ascensor por parte de una persona.



1.6 Opciones de representación.

Los argumentos de los mensajes se representan en los diagramas, ya sea utilizando pseudocódigo, o bien directamente en la sintaxis del lenguaje de programación. En lugar de emplear esas expresiones textuales para los argumentos y valores proporcionados, la notación propone también una representación gráfica de los argumentos en forma de flechas terminadas por pequeños círculos. El siguiente diagrama da un ejemplo de representación gráfica de argumentos de un mensaje.



Un símbolo es un circulito etiquetado con la expresión de argumento o con el nombre de valor proporcionado. Posee una flecha pequeña apuntando en el sentido del mensaje (cuando es un argumento) o en dirección contraria al mensaje (para un valor proporcionado) .

1.7 Expresiones en metamodelo

1.7.1 Las colaboraciones

Una colaboración es un mecanismo compuesto de elementos estructurales y de comportamiento. Las colaboraciones proporcionan un mecanismo de organización pero poseen, contrariamente a los paquetes, una identidad y un ámbito semántico. Un mismo elemento puede intervenir en varias colaboraciones.

Una colaboración engloba dos géneros de construcción: un contexto compuesto de una descripción de la estructura estática de los objetos afectados y una interacción representada por una secuencia de mensajes intercambiados por dichos objetos. Se requieren los dos aspectos para documentar plenamente el comportamiento, pero cada aspecto puede visualizarse independientemente.

Las colaboraciones emplean según el nivel de detalle, para describir especificaciones y para expresar realizaciones. La siguiente tabla resume los elementos de modelado que pueden ser descriptos por una colaboración.

Especificación	Tipo	Operación	Caso de uso
Realización	Clase	Método	Realización caso de uso

Las colaboraciones existen también bajo la forma genérica (modelo), parametrizada por clases, relaciones, atributos y operaciones. Una colaboración genérica se llama *pattern*, o esquema y microarquitectura. Los patterns poseen siempre un nombre, contrariamente a las colaboraciones que pueden ser anónimas.

1.7.2 Las interacciones

Una interacción expresa el comportamiento que resulta de la colaboración de un grupo de instancias. Una interacción puede visualizarse según el punto de vista del tiempo (por los diagramas de secuencia) o según el punto de vista del espacio (por los diagramas de comunicación). Las interacciones comprenden principalmente los siguientes elementos:

- las *instancias* que son la manifestación concreta de un tipo;
- los *enlaces* que enlazan las instancias y sirven de soporte para los envíos de mensaje;
- los *mensajes* que desencadenan las operaciones;
- las *funciones* implementadas por los extremos de los mensajes.

1.7.3 Rol de asociación:

Es una conexión de dos roles de clasificador dentro de una colaboración. Asociación entre dos clasificadores que se aplica sólo a un cierto contexto especificado por una condición.

Es una asociación que está definida y tiene significado sólo en el contexto descrito por una colaboración. Los roles de asociación son parte estructural clave de las colaboraciones, pues permiten describir relaciones contextuales. Dentro de una colaboración, un rol de clasificador denota una aparición individual de un clasificador, distinta de otras apariciones del clasificador y distinta también de la propia declaración del clasificador.

Se representa mediante una ruta entre dos símbolos de rol de clasificador.

Notación:

Nombre-Rol-Asociación Nombre-asociación-base

1.7.4 Rol de clasificador:

Es ranura de una colaboración que describe el rol desempeñado por un participante en una colaboración.

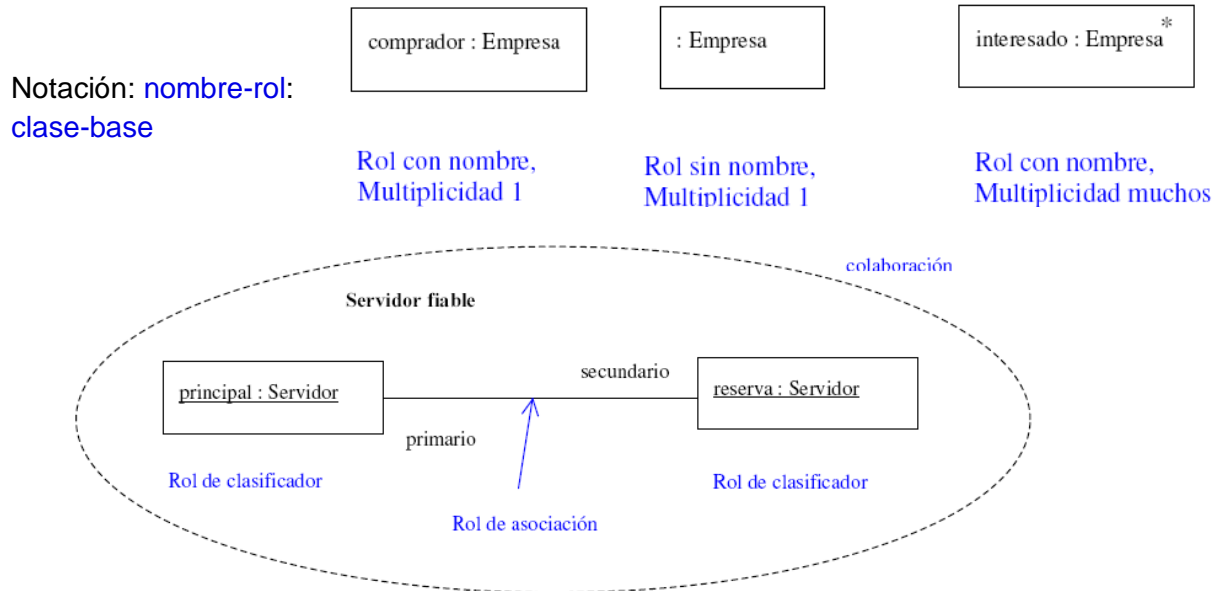
Una colaboración describe un patrón de interacción entre un conjunto de participantes, que son instancias de clases o de tipos de datos. Un rol de clasificador es la descripción de un participante. Cada rol es un uso diferente del clasificador en un contexto propio y único. Puede haber más de un rol para el mismo clasificador, cada uno de los cuales tendrá un conjunto diferente de relaciones con otros roles dentro de una colaboración. Sin embargo, un rol no es un objeto individual, sino una descripción de todos los objetos que pueden tomar parte en una instancia de una colaboración. En cada instancia de la colaboración, los roles serán desempeñados por distintos objetos y enlaces.

Un rol de clasificador tiene una referencia a un clasificador (clasificador base) y una multiplicidad. El clasificador base restringe el tipo de objeto que puede desempeñar el rol de clasificador. La clase del objeto puede ser la misma o un descendiente del clasificador base. La multiplicidad indica cuántos objetos pueden desempeñar el rol a la vez en una instancia de la

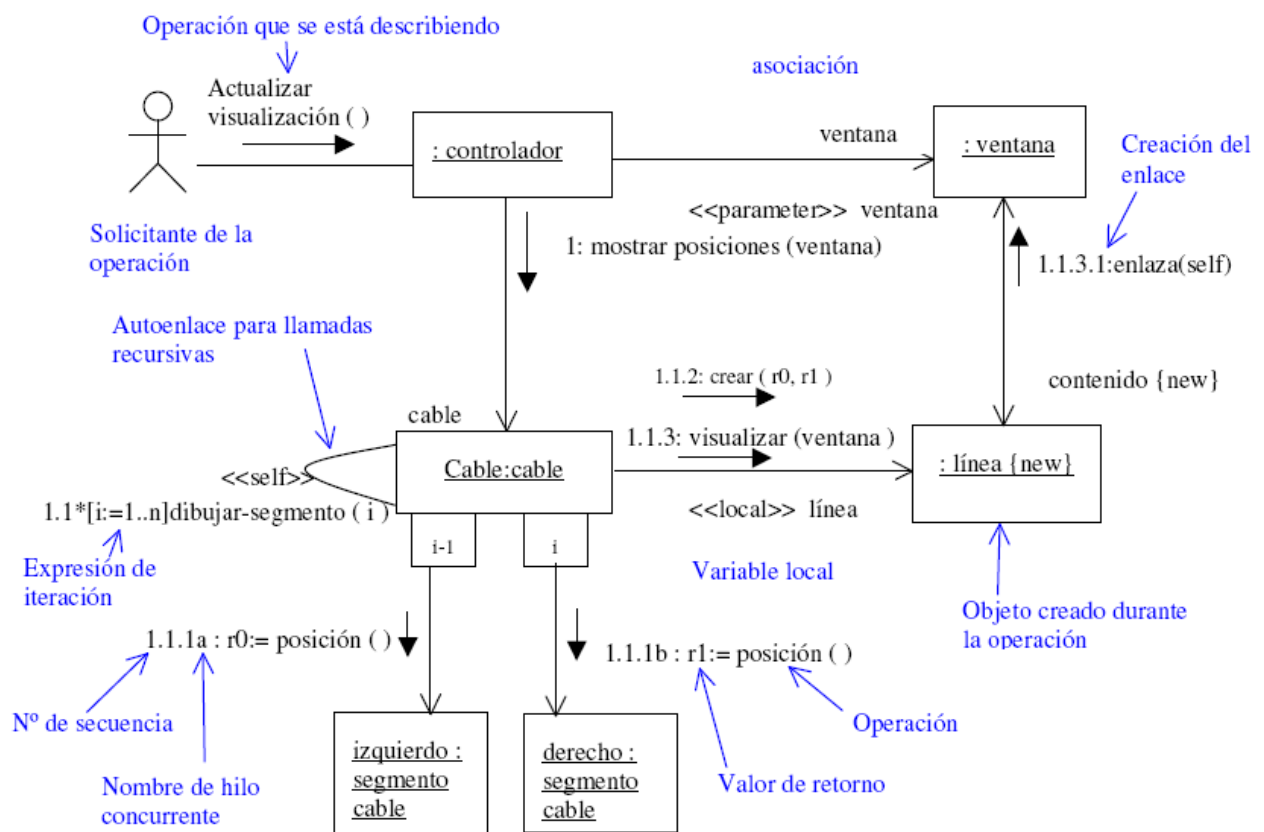
colaboración.

El mismo objeto puede desempeñar diferentes roles en distintas colaboraciones. Una colaboración representa una faceta del objeto.

Por ejemplo:



Cuadro ejemplo resumen:



Versión de UML 2.0

Diagramas De Comunicación.

Un diagrama de la comunicación, antes llamado diagrama de colaboración, es un diagrama de la interacción que muestra información similar a la de los diagramas de secuencia pero su foco primario se fija en las relaciones del objeto.

En los diagramas de comunicación, los objetos se muestran con los conectores de la asociación entre ellos. Los mensajes se agregan a las asociaciones como flechas que señalan en la dirección del flujo del mensaje. La secuencia de mensajes se muestra con un esquema de enumeración.

Los dos diagramas siguientes demuestran un diagrama de comunicación y el diagrama de secuencia que muestra la misma información. Aunque es posible derivar el orden de los mensajes en el diagrama de la comunicación mediante la enumeración, él no se visualiza inmediatamente.

