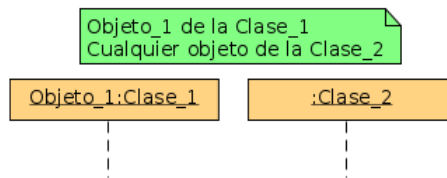


6.B. Diagramas de interacción.

2. Diagramas de secuencia.

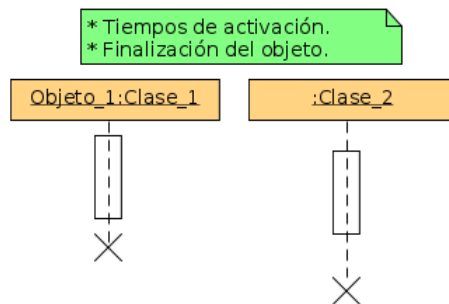
2.1. Representación de objetos, línea de vida y paso de mensajes.

En un **diagrama de secuencia**, los **objetos** se dibujan mediante rectángulos y se distribuyen horizontalmente en la parte superior del diagrama. Por cada objeto se identifica su nombre, seguido del símbolo de dos puntos y a continuación el nombre de la clase a la que pertenece. Si no se indica el nombre del objeto, se considera que para el propósito del diagrama es válido cualquier objeto de la clase.



De cada rectángulo sale una **línea punteada** que representa el paso del tiempo, se denomina **línea de vida**. La línea de vida se prolonga mientras el objeto es relevante en el diagrama, una vez deja de serlo se indica mediante una cruz "X", dejando por tanto de existir a partir de ese momento.

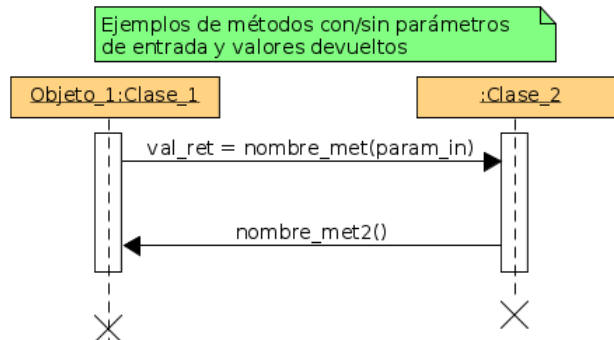
Cuando el **objeto** toma **protagonismo** en el **intercambio** de mensajes, se dice que está **activo** y se indica mediante un recuadro sobre su línea de vida.



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.

Invocación de métodos.

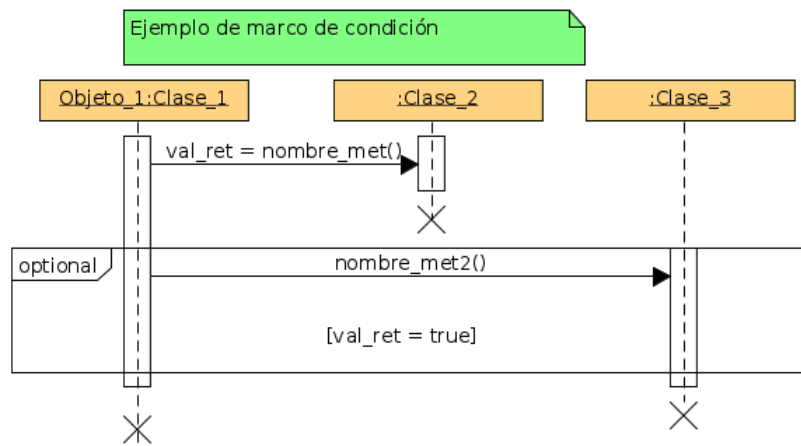
Los **mensajes**, que significan la invocación de métodos, se representan como flechas horizontales que van de una línea de vida a otra, indicando con la flecha la dirección del mensaje. Los mensajes se dibujan desde el objeto que envía el mensaje al que lo recibe, pudiendo ser el mismo objeto emisor y receptor de un mensaje. El orden en el tiempo 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. Los **mensajes** tendrán un **nombre** y pueden incluir **argumentos de entrada**, **valores devueltos** e **información de control** (condición o iteración).



Una notación alternativa para recoger valores devueltos por los métodos es dibujar una **línea de puntos** finalizada en flecha, que irá desde el objeto destinatario del mensaje al que lo ha generado, acompañado del texto del valor devuelto.

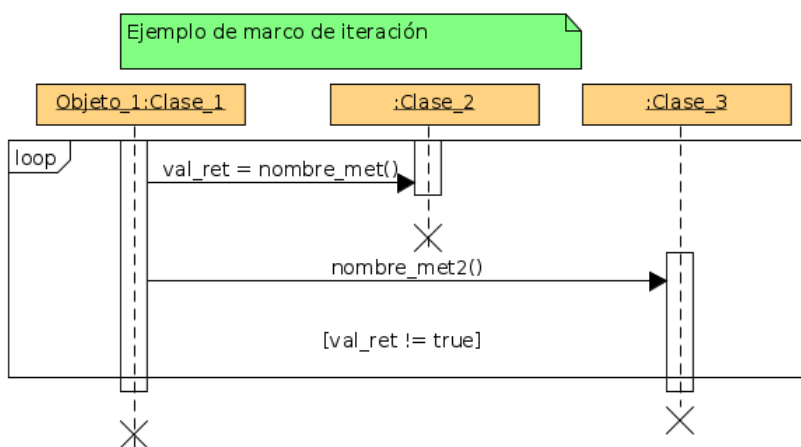
Las secuencias de control; tanto **condicionales**, como **iterativas**, se pueden representar usando **marcos**, normalmente se nombra el marco con el tipo de bucle a ejecutar y la condición de parada. También se pueden representar flujos de mensajes condicionales en función de un valor determinado.

La expresión a evaluar para la condición o iteración se representa entre corchetes.



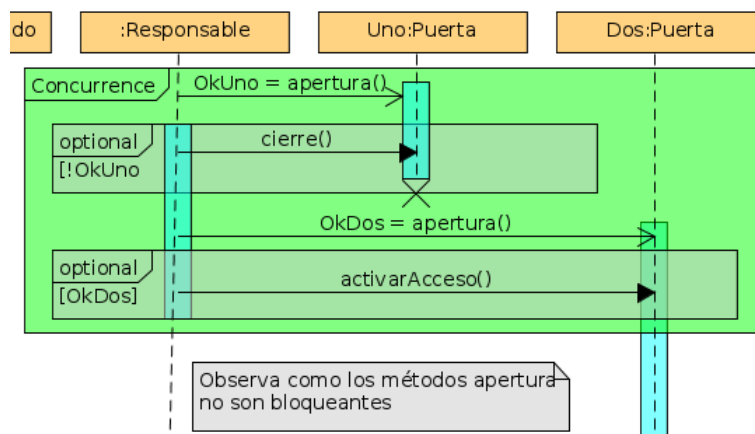
Combinando varios marcos opcionales es posible representar diferentes alternativas en la ejecución de un diagrama de secuencia.

Para el caso de una iteración, tenemos el siguiente ejemplo.



Por defecto los métodos son **bloqueantes**, se entiende que el proceso del diagrama de secuencia completa cada método antes de continuar con el siguiente, es una secuencia de métodos en el tiempo. Pero en ocasiones se producen situaciones en las que se desea mostrar varios procesos en paralelo (**conurrencia**), se puede reflejar mediante el uso de **marcos con la etiqueta concurrence**.

Junto a los marcos de concurrencia, se hace necesario el uso de **métodos no bloqueantes (asíncronos)**, que permitan en paralelo activar diferentes procesos. La notación utilizada para los **métodos asíncronos** es una línea finalizada con **media cabeza de flecha** o en UMLet una línea cuya punta flecha no está rellena.



Por último, indicar que hay situaciones difíciles de reflejar en un diagrama de secuencia, en estas ocasiones el uso de **etiquetas y notas** resulta de suma utilidad.