

# Modelado de un Sistema con Diagramas UML

**Programacion Concurrente  
2025**

# Introducción UML:

- UML: **lenguaje unificado de modelado**
- Es un lenguaje gráfico para modelar, visualizar, especificar, construir y documentar un sistema de software.
- Es importante remarcar que UML es un "lenguaje de modelado" para *especificar o para describir* métodos o procesos.
- Se utiliza para definir un sistema, para detallar las partes del mismo y para documentar y construirlo. En otras palabras, es el lenguaje en el que está descripto el modelo del sistema.
- UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos/vistas de las entidades representadas del sistema.

- Tipos de Diagramas:
  - **Estructurales**
    - Muestran la estructura estática de los objetos en un sistema por ej.
      - ***Diagrama de Clases.***
      - *Diagrama de componentes.*
      - Diagramas de despliegue.
      - *Diagrama de objetos.*
      - Diagrama de paquetes.
      - Diagrama de perfiles.
      - Diagrama de estructura compuesta.

- Tipos de Diagramas:
  - **De Interacción y Comportamiento:**
    - Muestran el comportamiento dinámico de los objetos en el sistema por ej.
      - ***Diagrama de Secuencia.***
      - Diagrama de Actividades.
      - *Diagrama de Casos de Uso.*
      - Diagrama global de interacciones.
      - Diagrama de comunicación.
      - Diagrama de tiempos.

# UML – Diag. Estructural

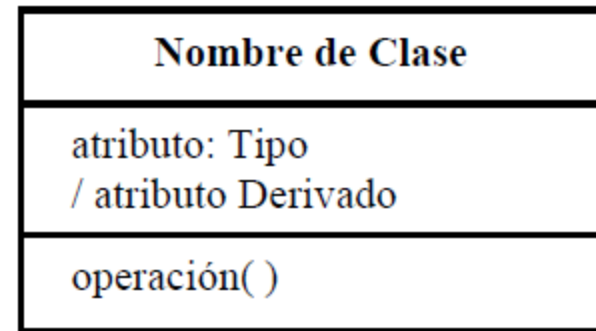
## Diagramas de Clases

- Permite visualizar las relaciones entre las clases del sistema.
- Los diagramas de clases son el tipo de diagrama UML de estructura estática **más utilizado**.
- Es el bloque de construcción principal de cualquier solución orientada a objetos.
- En un diagrama de clases se muestra
  - Las clases del sistema.
  - Nombre y atributos de cada clase.
  - Operaciones de cada clase.
  - Relación entre cada clase.

# UML – Diag. Estructural

## Diagramas de Clases

- En la mayoría de las herramientas de modelado, una clase tiene tres partes, nombre en la parte superior, atributos en el centro y operaciones o métodos en la parte inferior.



- En sistemas grandes con muchas clases relacionadas, las clases se agrupan.
- Las Diferentes relaciones entre las clases se muestran por diferentes tipos de flechas.

# UML – Diag. Estructural

## Diagramas de Clases

- En los diagramas de clase se puede describir la **visibilidad** de los miembros de la clase (atributo o método).
  - Public: el miembro es visible dentro y fuera de la clase. +
  - Private: el miembro es accesible solo desde adentro de la clase. -
  - Protected: el miembro no será accesible desde fuera de la clase, pero si podrá ser accesado por métodos de la clase además de las subclases que se deriven (herencia). #


# UML – Diag. Estructural

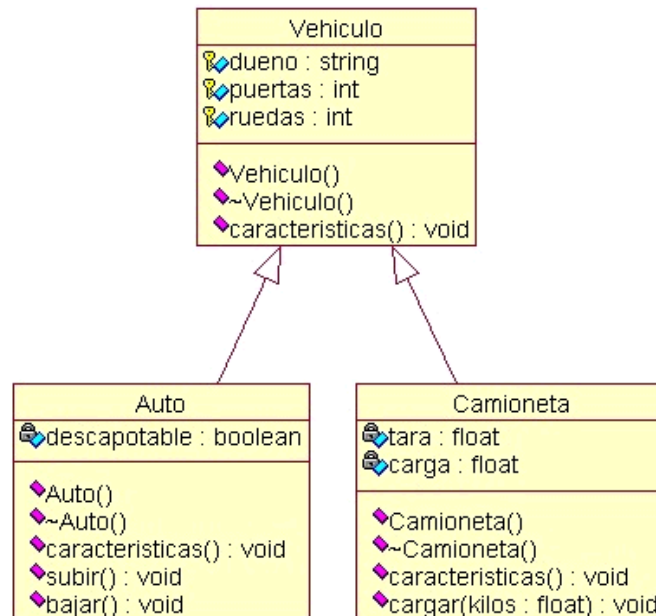
## Diagramas de Clases

- En los diagramas de clase se puede describir la relación entre las clases.
  - **uno o muchos:**  $1..* (1..n)$
  - **0 o muchos:**  $0..* (0..n)$
  - **número fijo:**  $m$  ( $m$  denota el número).
- Además puede describirse relaciones a nivel de instancia:
  - Herencia - Asociación – (Agregación – Composición)
  - Uso




# UML – Diag. Estructural

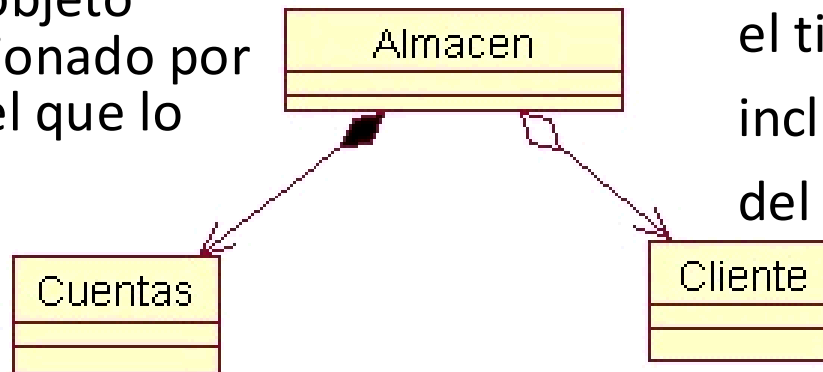
- Relaciones a nivel de instancia:
  - Herencia 
  - Una subclase hereda atributos y métodos de la superclase (public y protected).



# UML – Diag. Estructural


- Relaciones a nivel de instancia:
  - Agregación 
  - Sirve para el modelado de sistemas complejos.

- Composición (ej cuentas)  
Relación estática, en donde el tiempo de vida del objeto incluido esta condicionado por el tiempo de vida del que lo incluye



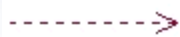
- Agregación (ej cliente)  
relación dinámica, en donde el tiempo de vida del objeto incluido es independiente del que lo incluye

# UML – Diag. Estructural

- Relaciones a nivel de instancia:
  - Asociación 
    - Permite asociar objetos que colaboran entre si
    - No es una relación fuerte, es decir, el tiempo de vida de un objeto no depende del otro.
    - Si la asociación es entre dos clases se puede representar con una línea.



# UML – Diag. Estructural

- Relaciones a nivel de instancia:
  - Instanciación o Uso: 
  - Utilizada para denotar la instanciación y dependencia de una clase a otra.
  - Por ej. Una aplicación gráfica que instancia una ventana



# UML Diag. de Interacción

## Diagramas de Secuencia

- Los Diagramas de Secuencias muestran la forma en que ***un grupo de objetos*** se ***comunican*** (interactúan) entre sí a lo largo del ***tiempo***
- Permite modelar cómo los objetos interactúan y el orden en que se producen esas interacciones.
- Se debe modelar la interacción de los objetos del sistema para cada caso de uso.

# UML Diag. de Interacción

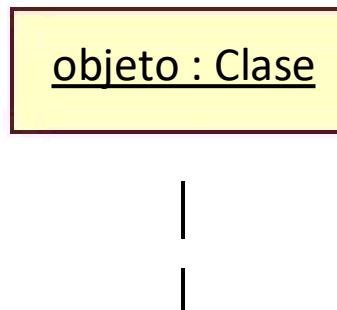
## Diagramas de Secuencia

- Un Diagrama de secuencia consta de objetos, mensajes entre estos objetos y una línea de vida del objeto representada por una línea vertical.
- El tiempo transcurre de arriba hacia abajo.

# UML Diag. de Interacción

## Diagramas de Secuencia - Componentes

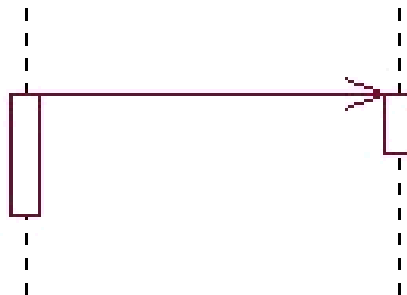
- Objeto/Actor. Representado por un rectángulo, modela una instancia de un objeto en particular.



# UML Diag. de Interacción

## Diagramas de Secuencia - Componentes

- Invocación de un método de otro objeto.
  - Se representa por una flecha entre un objeto y el invocado, representa la llamada de un método (operación) de un objeto en particular.



- Si existe demora entre el envío y la atención puede indicarse con una línea oblicua.
- El metodo invocado retorna un valor, que se representa con una linea punteada terminada en flecha.



# UML Diag. de Interacción

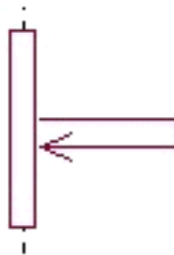
## Diagramas de Secuencia - Componentes

- Mensaje de retorno.
  - Se representa por una flecha entre un objeto y otro (generalmente en línea discontinua)
  - Si existe demora entre el envío y la atención puede indicarse con una línea oblicua.

# UML Diag. de Interacción

## Diagramas de Secuencia - Componentes

- Mensaje a mismo objeto.
  - Se representa por una flecha al mismo objeto indicando el **llamado a un método propio**.



# UML Diag. de Interacción

- Fragmento Combinado:
  - Permite representar la interacción entre fragmentos.
  - Un fragmento combinado se define por un operador interacción y los operandos de interacción correspondientes.
  - Permite representar trazas de manera compacta y concisa.
  - Pueden tener limitaciones de interacción llamados guardas UML 2.4

# UML Diag. de Interacción

- Operadores de interacción:

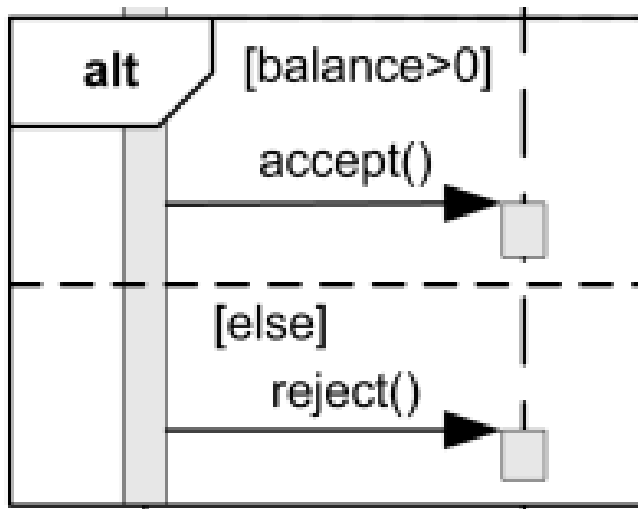
- alt - [alternatives](#)
- opt - [option](#)
- loop - [iteration](#)
- break - [break](#)
- par - [parallel](#)
- strict - [strict sequencing](#)
- seq - [weak sequencing](#)
- critical - [critical region](#)
- ignore - [ignore](#)
- consider - [consider](#)
- assert - [assertion](#)
- neg - [negative](#)

# Interaction Constraint

- Para hacer una **restricción en una interacción** se usa una expresión booleana en la **guarda**
  - operando combinado.
- Una restricción se muestra en la interacción entre corchetes que cubre la **línea de vida** .
- UML 2.4 a menudo se refiere a la restricción de la interacción como un **guarda** .

# Alternativas

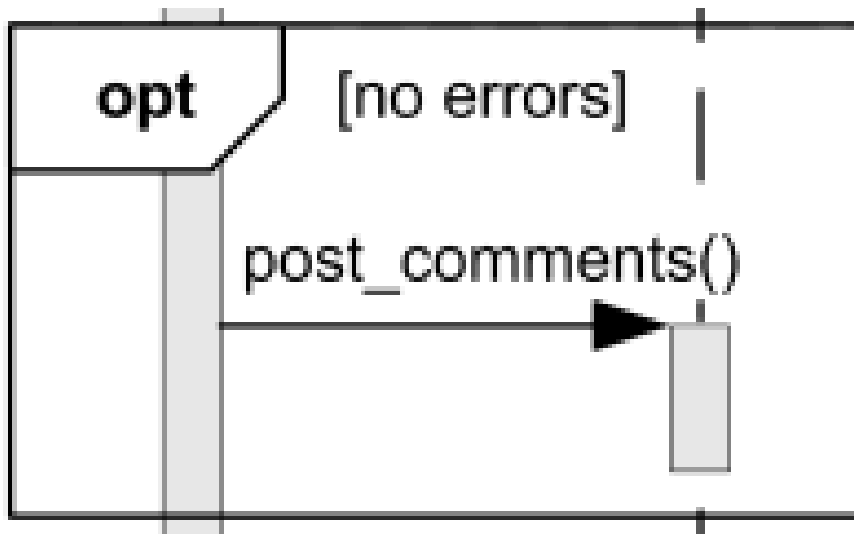
- El operador de interacción **alt** representa una **elección** o alternativas de comportamiento.
  - Solo uno de los operando se elegirá.
  - El operando elegido debe tener una expresión de guarda explícita o implícita que se evalúa como verdadera en este momento de la interacción.
  - Un verdadero es implícito si el operando no tiene guarda.



```
if (condicion){  
}  
else{  
}
```

# Opción

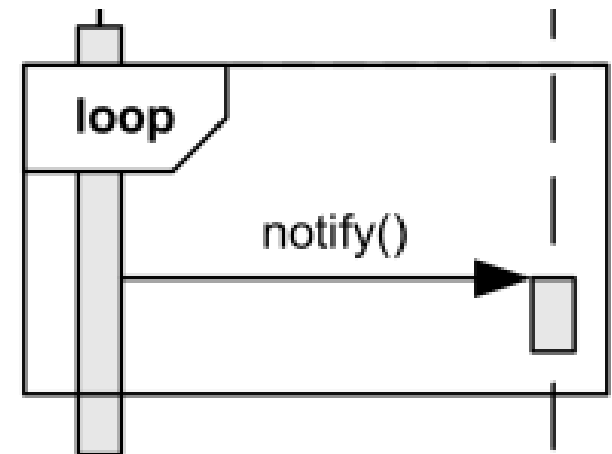
- El operador de la interacción **opt** significa que el fragmento combinado representa una **elección** de la conducta, ya sea que una única operando ocurra o no ocurre nada.
- Una opción es semánticamente equivalente a un fragmento combinado alternativo donde hay un operando con un contenido que no esté vacía y el segundo operando es vacía



```
if (condicion){  
}  
}
```

# Loop

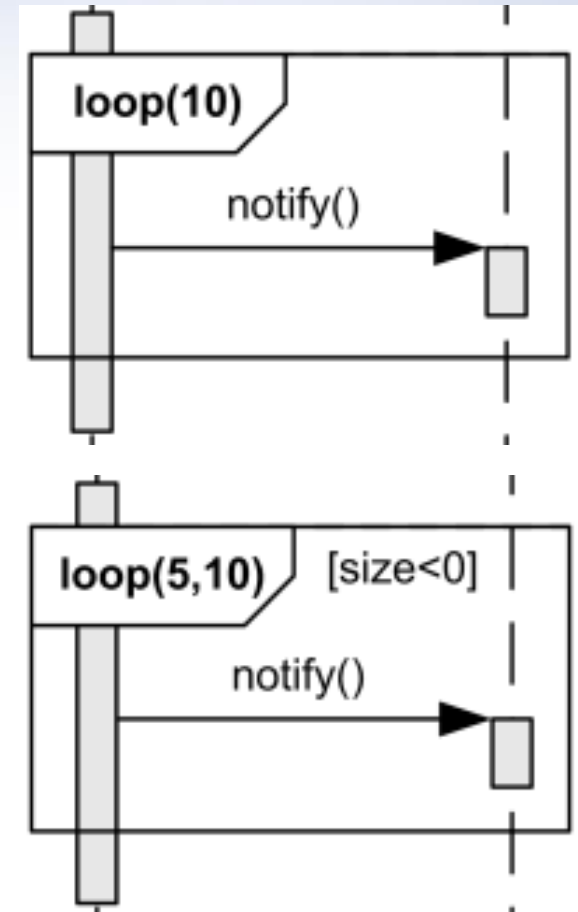
- El operador de la interacción **del bucle** significa que el fragmento combinado representa un bucle.
  - El operando de bucle se repite un número de veces.
- Loop podría ser controlado por uno o los dos límites de iteración y una guarda.
  - Si bucle no tiene límites especificados, significa bucle infinito potencial con el cero como límite inferior y límite superior infinita.





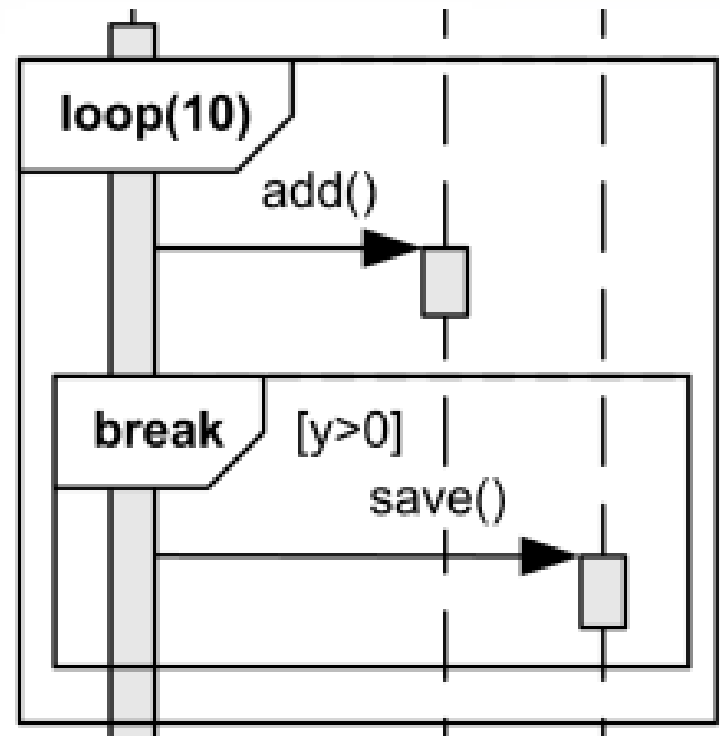
# Loop

- *Loop para ejecutar exactamente 10 veces.*
- *Podemos suponer que según UML 2.3, se espera que el bucle para ejecutarse mínimo de 5 veces y no más de 10 veces. Si la condición de guardia [tamaño < 0] se convierte en bucle falsa termina sin importar el número mínimo de iteraciones especificado.*



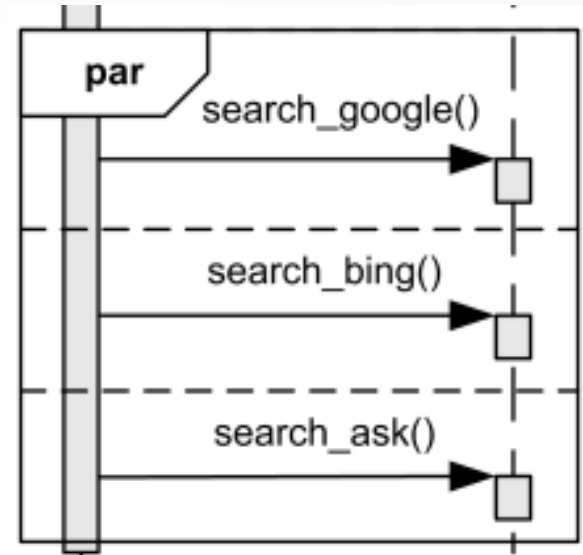
# Break

- El operador de la interacción **ruptura** representa una **ruptura en un** escenario o excepcional que se realiza en lugar de el resto del fragmento de interacción que encierra.
  - Un operador **Break con** una **guarda** se elige cuando la guarda es cierta.
  - En este caso se ignora el resto del fragmento de interacción que encierra directamente.
- Cuando la guarda del operando break es falsa, el operando se ignora y el resto de las sentencias que encierran fragmentos de interacción.



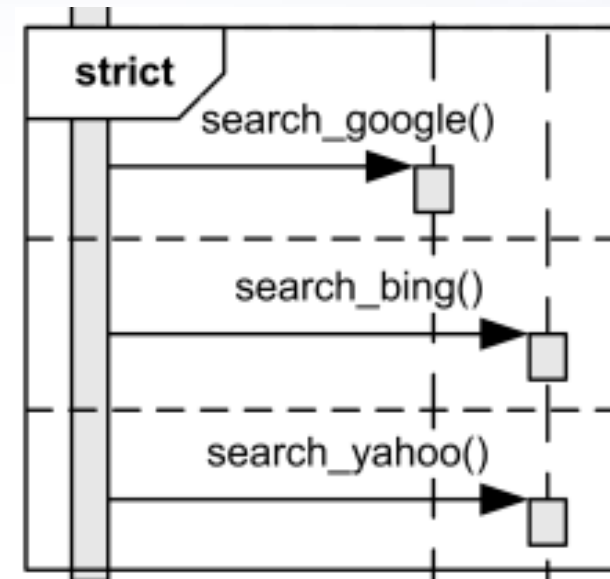
# Parallel

- El operador de la interacción **par** define la ejecución de ejecuciones potencialmente paralelas.
- Diferentes operandos pueden ser intercalados en cualquier forma, siempre y cuando el orden impuesto por cada operando se conserve.



# Strict Sequencing

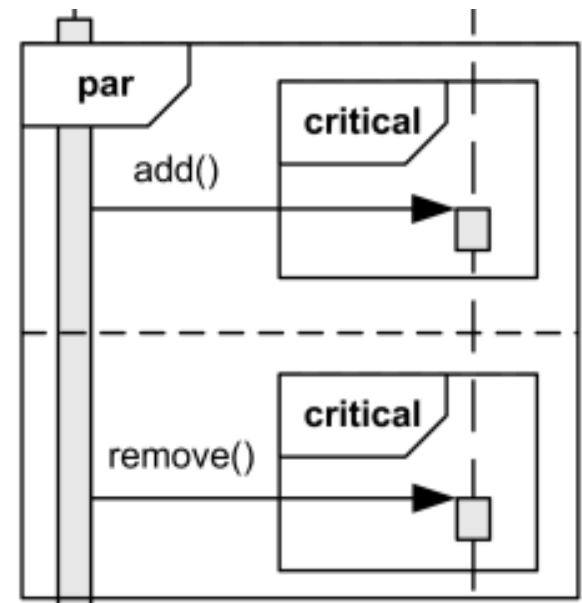
- El operador de la interacción **estricta** requiere una estricta secuencia (orden) de los operandos en el primer nivel dentro del fragmento combinado.
- Operandos de los niveles más bajos dentro del fragmento combinado contenida no se pueden comparar directamente con otras especificaciones de la ocurrencia del fragmento combinado que encierra.
- La notación, esto significa que la coordenada vertical de los fragmentos contenidos es significativo en todo el alcance del fragmento combinado y no sólo en una línea de vida.



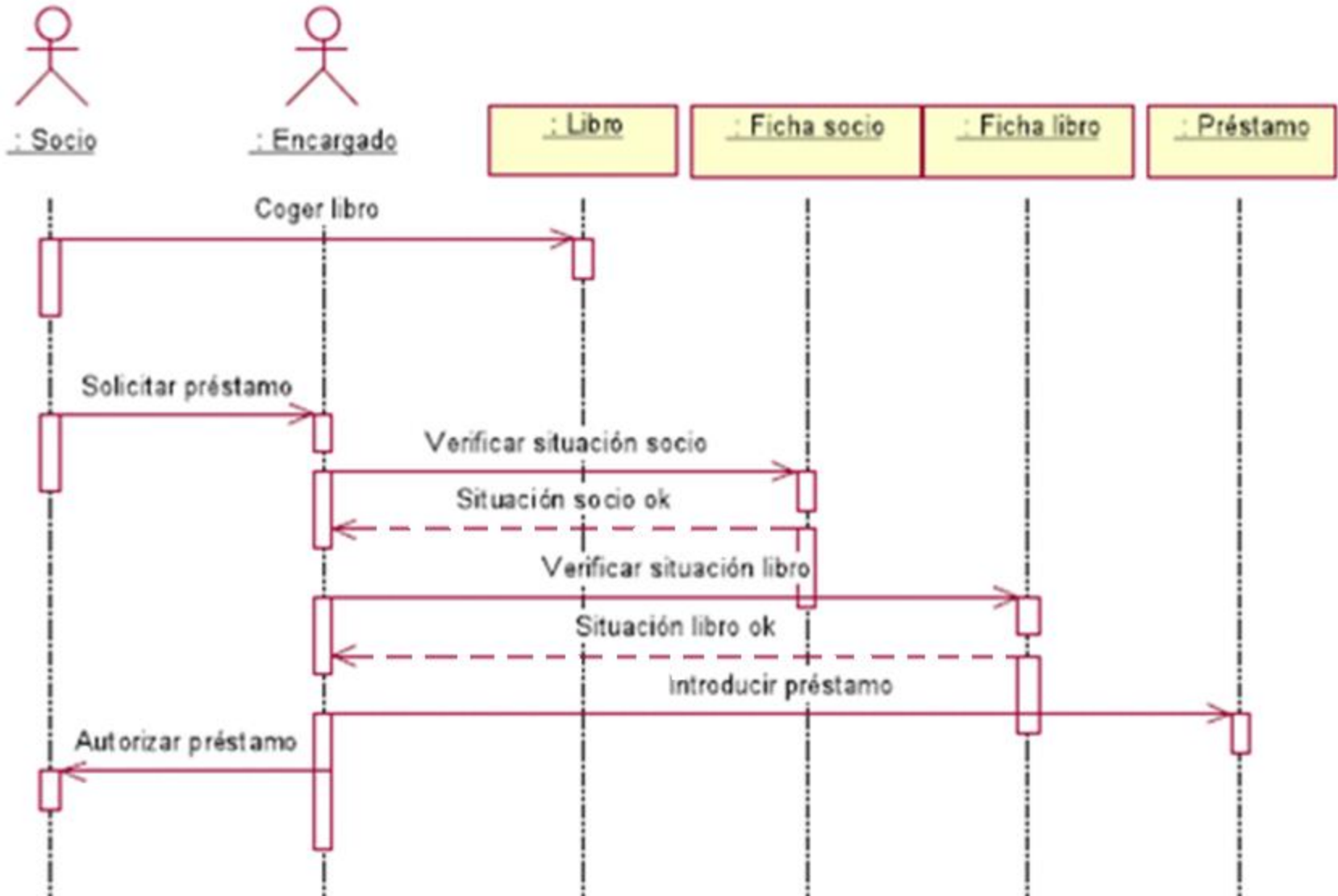
# Critical Region

- El operador interacción **crítica** define que el fragmento combinado representa una **región crítica**.
- Una región crítica es una región que no pueden ser intercaladas por otras especificaciones de ocurrencia (en las líneas de vida cubiertos por la región).
- Esto significa que la región se trata **atómicamente** por el fragmento que encierra y no puede ser intercalada, por ejemplo, por operador paralelo.

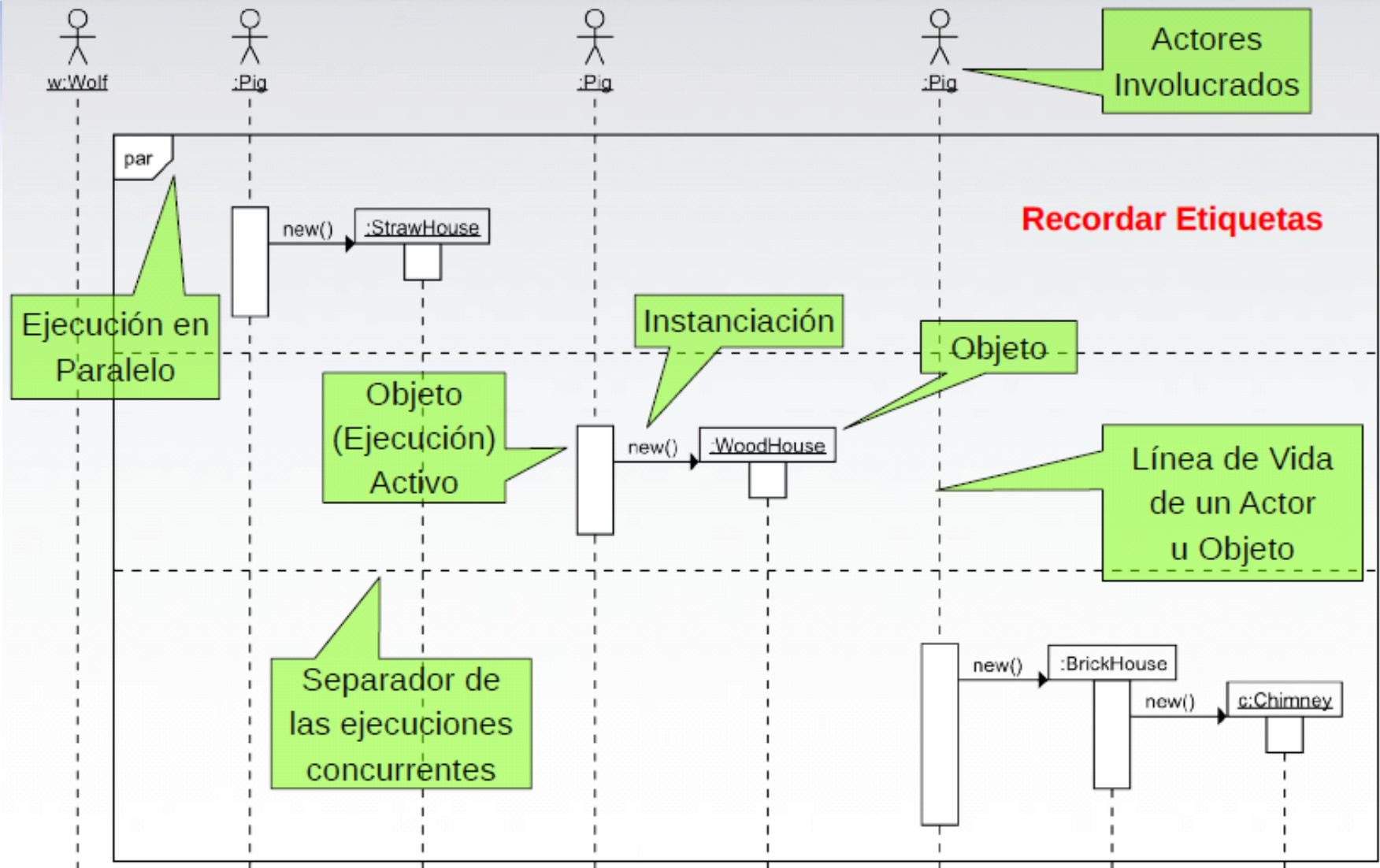
*Agregar () o eliminar () podría ser llamado en paralelo, pero cada uno debe funcionar como una región crítica.*



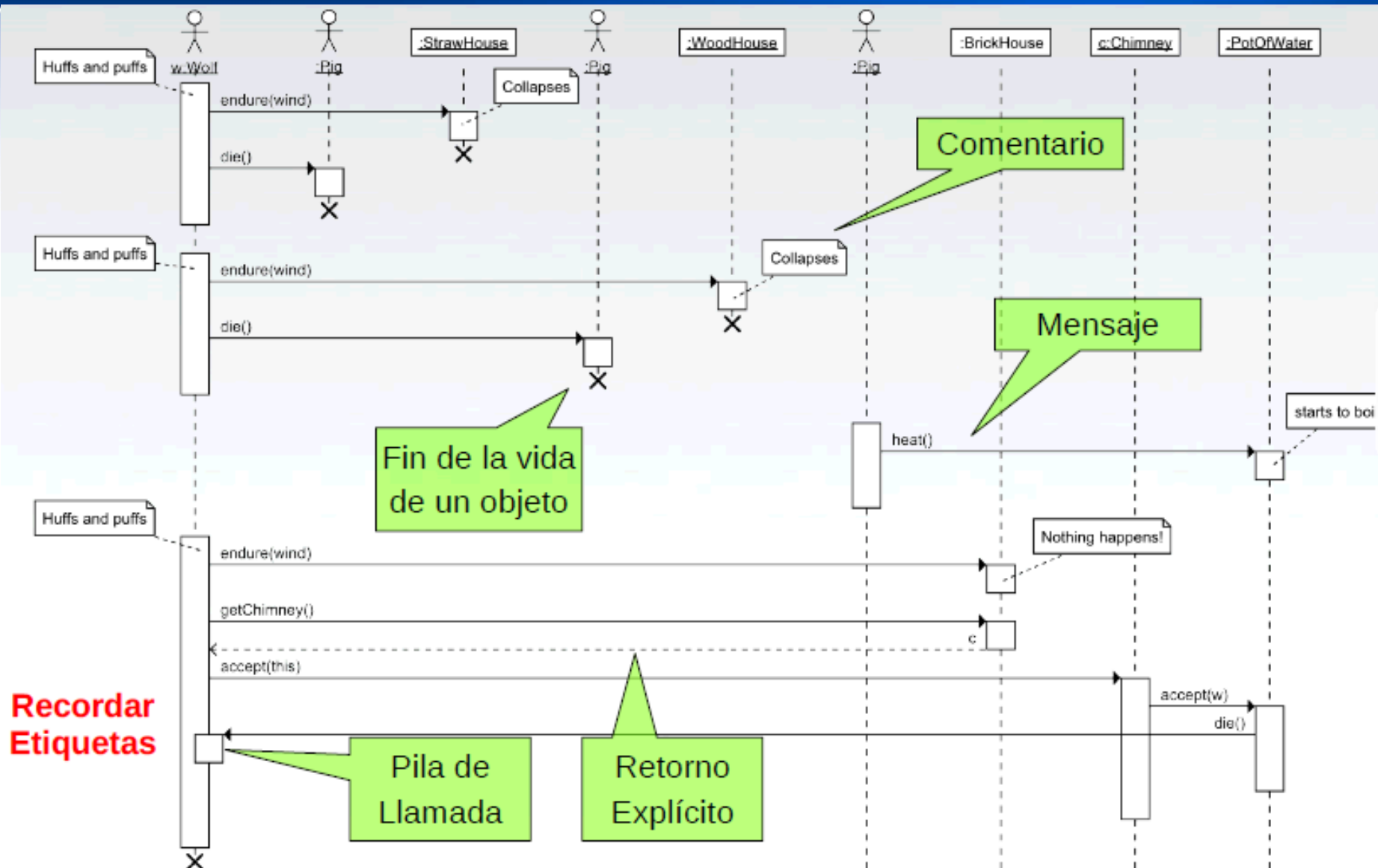
# Diag. de Secuencia Ej 1



# Diag. de Secuencia Ej 2a



# Diag. de Secuencia Ej 2b





# UML Diag. de Secuencia

Tutorial de diagrama de secuencias.

[http://www.tracemodeler.com/articles/a quick introduction to uml sequence diagrams/index.html](http://www.tracemodeler.com/articles/a_quick_introduction_to_uml_sequence_diagrams/index.html)

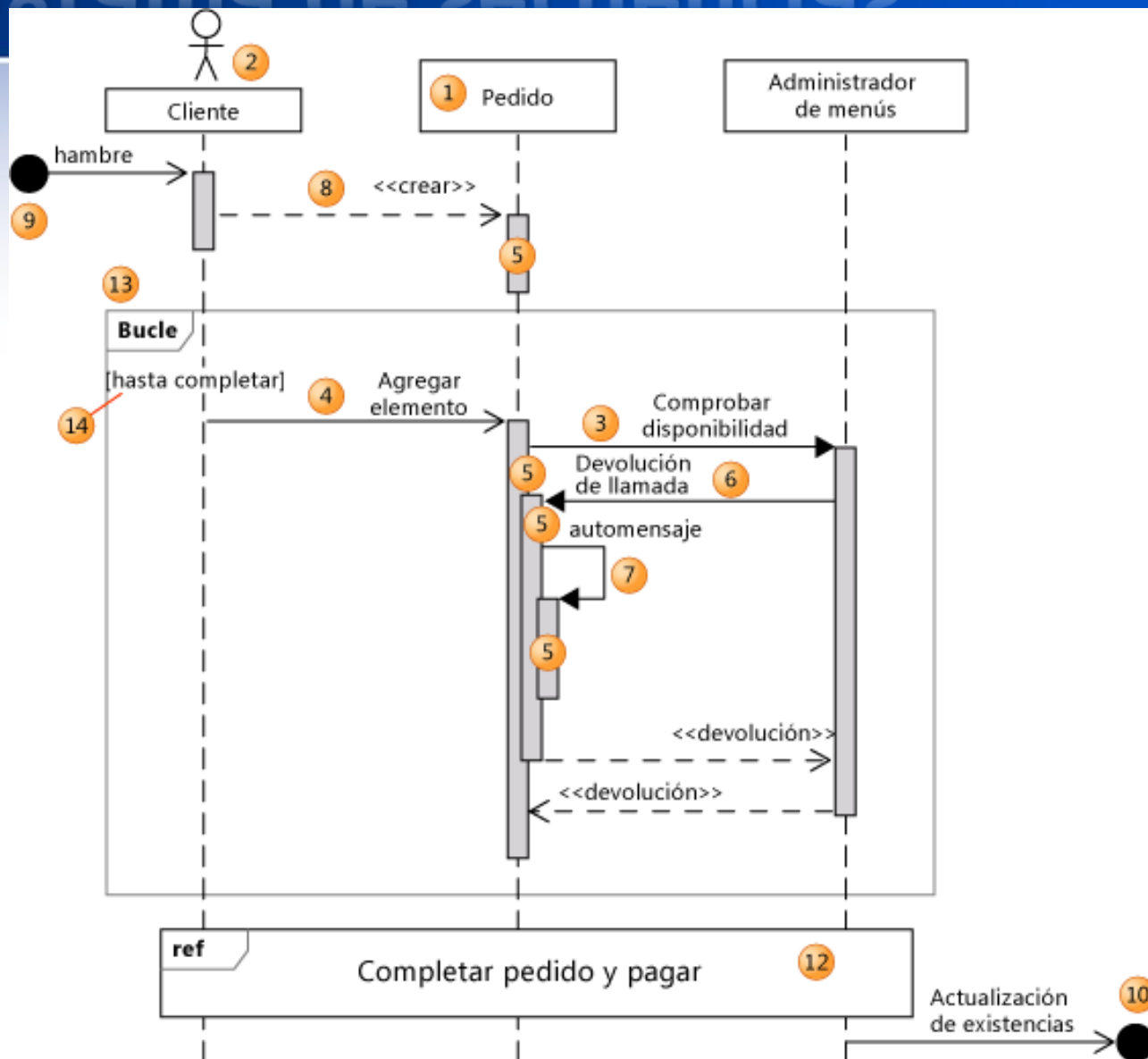
Herramientas interesantes:

- Visual Paradigm
- StarUML.
- Trace Modeler.
- Papyrus.

# UML Diag. de Secuencia

Apéndice. Ejemplo de partes de un diagrama de secuencia.

# Introduction



Forma	Elemento	Descripción
1	Lifeline	Línea vertical que representa la secuencia de eventos que se producen en un participante durante una interacción, mientras el tiempo avanza por la línea. Este participante puede ser una instancia de una clase, un componente o un actor.
2	Actor	Participante externo al sistema que está desarrollando. Para que aparezca un símbolo de actor al principio de una línea de vida, establezca la propiedad Actor.
3	Mensaje sincrónico	El remitente espera una respuesta a un mensaje sincrónico antes de continuar. El diagrama muestra la llamada y la devolución. Los mensajes sincrónicos se usan para representar llamadas de función ordinarias dentro de un programa, así como otros tipos de mensaje que se comportan de la misma manera.
4	Mensaje asincrónico	Mensaje que no requiere una respuesta para que el remitente continúe. Un mensaje asincrónico muestra solo una llamada del remitente. Se usa para representar la comunicación entre subprocesos diferentes o la creación de un nuevo subproceso.
5	Ocurrencia de ejecución	Rectángulo sombreado vertical que aparece en la línea de vida de un participante y representa el período en el que el participante ejecuta una operación. La ejecución empieza cuando el participante recibe un mensaje. Si el mensaje de inicio es un mensaje sincrónico, la ejecución finalizará con una flecha de retorno al remitente.
6	Mensaje de devolución de llamada	Mensaje que se devuelve a un participante que espera la devolución de una llamada anterior. La ocurrencia de ejecución resultante aparece encima de la que ya existe.
7	Mensaje propio	Mensaje de un participante a sí mismo. La ocurrencia de ejecución resultante aparece encima de la ejecución de envío.
8	Mensaje de creación	Mensaje que crea un participante. Si un participante recibe un mensaje de creación, este debe ser el primero que reciba.

Forma	Elemento	Descripción
9	Mensaje encontrado	Mensaje asincrónico de un participante desconocido o no especificado.
10	Mensaje perdido	Mensaje asincrónico a un participante desconocido o no especificado.
11	Comentario	Se puede asociar un comentario a cualquier punto de una línea de vida.
12	Interaction Use	Contiene una secuencia de mensajes definidos en otro diagrama. Para crear un uso de interacción, haga clic en la herramienta y, después, arrastre el mouse por las líneas de vida que quiere incluir.
13	Fragmento combinado	Colección de fragmentos.Cada fragmento puede incluir uno o varios mensajes.Hay varios tipos de fragmentos combinados.Para más información, vea <a href="#">Describir el flujo de control con fragmentos de diagramas de secuencia de UML</a> . Para crear un fragmento, haga clic con el botón derecho en un mensaje, seleccione Delimitar con y, después, haga clic en un tipo de fragmento.
14	Restricción de fragmentos	Se puede usar para indicar una condición sobre si tendrá lugar el fragmento. Para establecer la restricción, seleccione un fragmento y, después, seleccione la restricción y escriba un valor.
X	Evento de destrucción	Representa el punto en el que el objeto se ha eliminado o ya no es accesible.Aparece en la parte inferior de cada línea de vida.
	Interacción	Colección de mensajes y líneas de vida que se muestra en el diagrama de secuencia.Para ver las propiedades de una interacción, debe seleccionarla en el Explorador de modelos UML.
	Diagrama de secuencia	Diagrama que muestra una interacción.Para ver sus propiedades, haga clic en una parte vacía del diagrama.