TEMA 7. DIAGRAMAS DE SECUENCIA

ANÁLISIS Y DISEÑO DE SISTEMAS DE INFORMACIÓN

Universidad Francisco de Vitoria
UFV Madrid

ÍNDICE

Tema 7: Diagramas de secuencia

1. INTRODUCCIÓN

Modelado del comportamiento entre objetos = **interacciones entre objetos**

Interacción. Especifica cómo se intercambian los mensajes y los datos entre los actores de interacción.

Actores

- Humanos/Personas (profesor, administrador, etc.)
- No humanos (servidor, impresora, software ejecutable, etc.)

Ejemplos de interacciones:

- Comunicación entre personas.
- Intercambio de mensajes entre personas y un sistema de software.
- Protocolos de comunicación.
- Secuencia de llamadas a métodos en un programa, etc.

2. DIAGRAMAS DE INTERACCIÓN

Se usa para especificar interacciones.

Modelado de escenarios concretos.

Describir **secuencias de comunicación** en diferentes niveles de detalle.

Los diagramas de interacción muestran:

- Interacción de un sistema con su entorno.
- Interacción entre las partes del sistema para mostrar cómo se puede implementar un caso de uso específico.
- Comunicación entre procesos en la que los actores involucrados deben cumplir ciertos protocolos.
- Comunicación a nivel de clase (llamadas/operaciones, comportamiento entre objetos)

3. DIAGRAMAS DE SECUENCIA

Diagrama bidimensional

- **Eje horizontal**: actores de interacción involucrados
- **Eje vertical**: el tiempo. Orden cronológico de la interacción

Interacción = secuencia de eventos

Una línea de vida muestra una participación individual en la interacción. Representa la existencia de un objeto.

Notación: rectángulo con una línea discontinua debajo. Una cruz al final indica la destrucción del objeto (evento de destrucción)

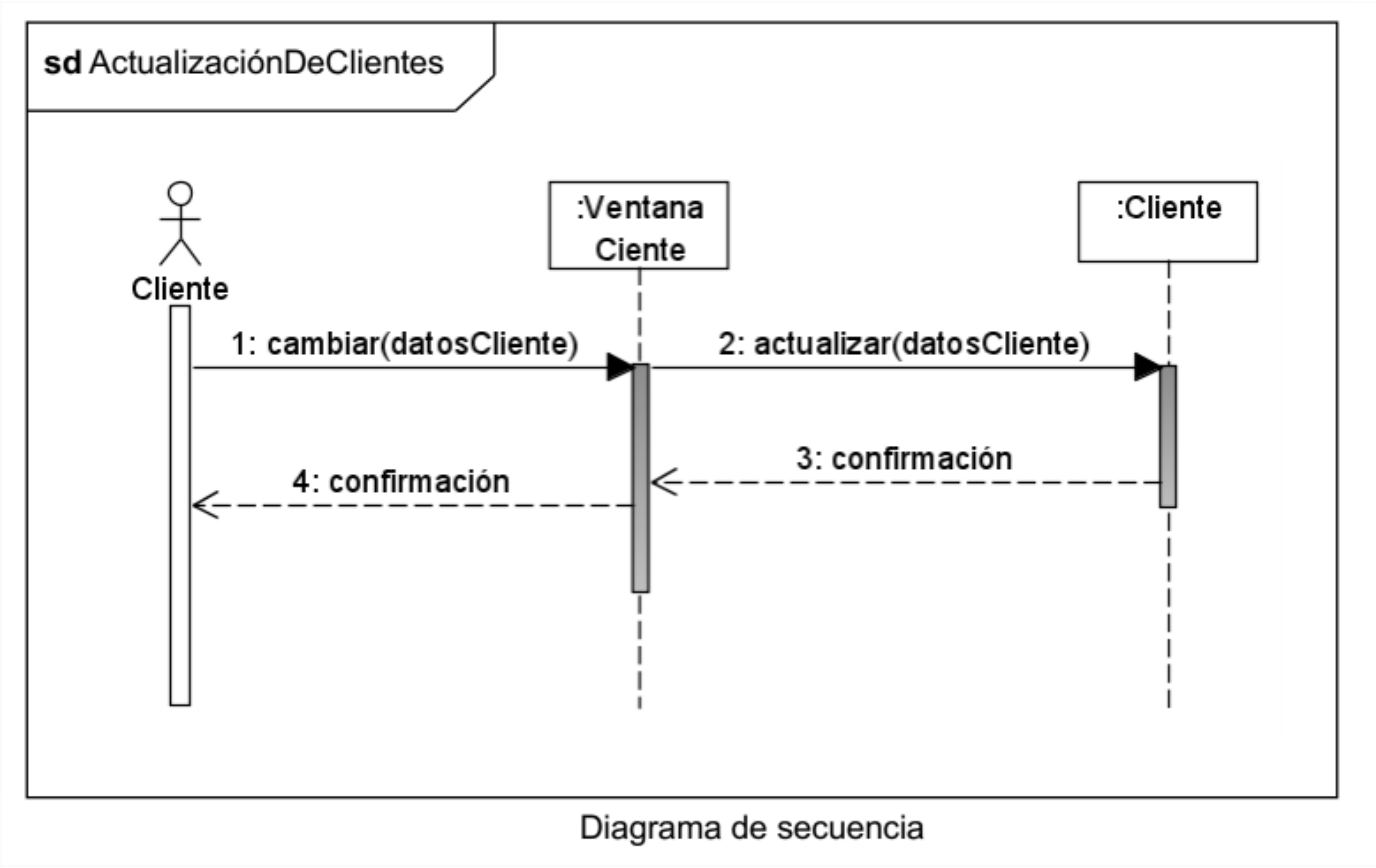
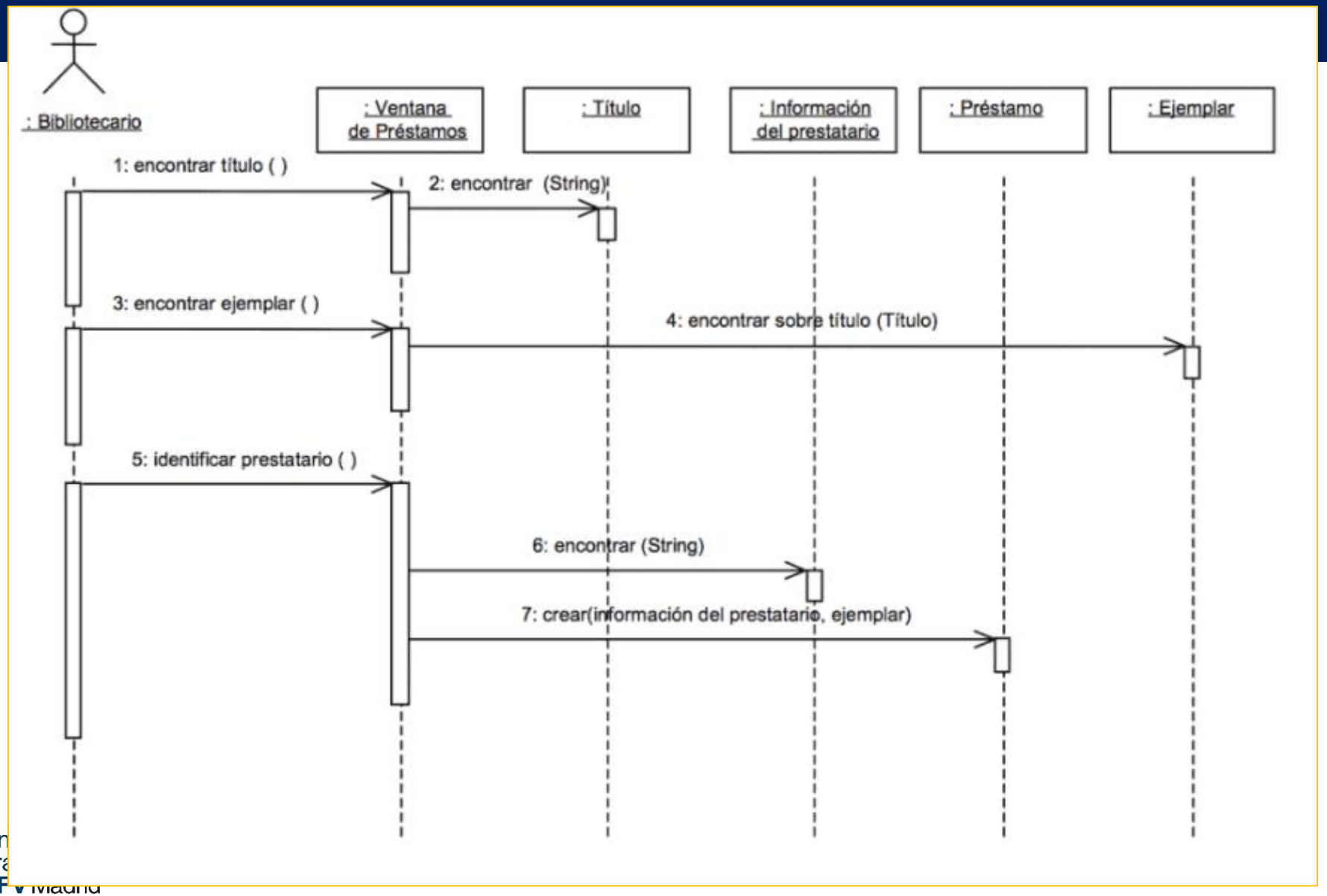
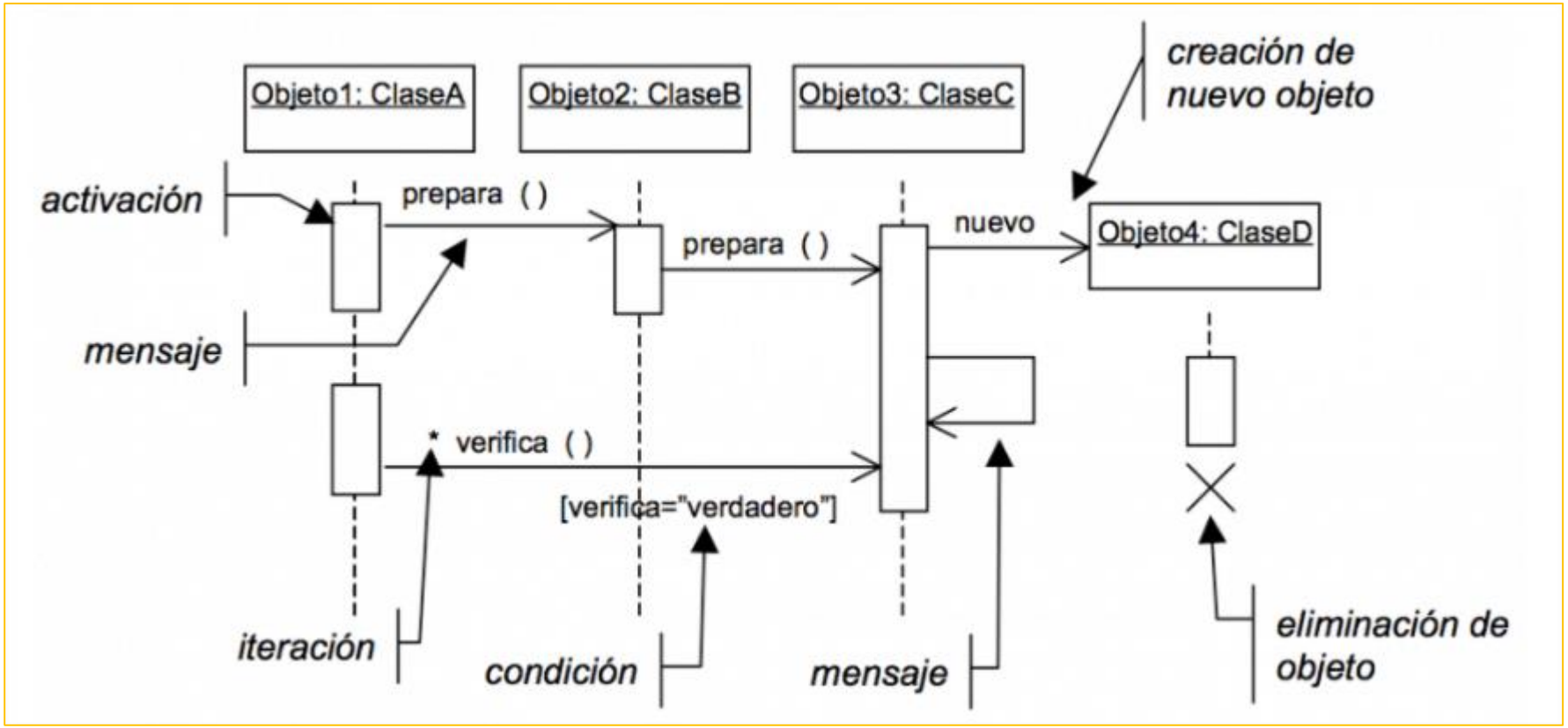


Diagrama de secuencia

3. DIAGRAMAS DE SECUENCIA

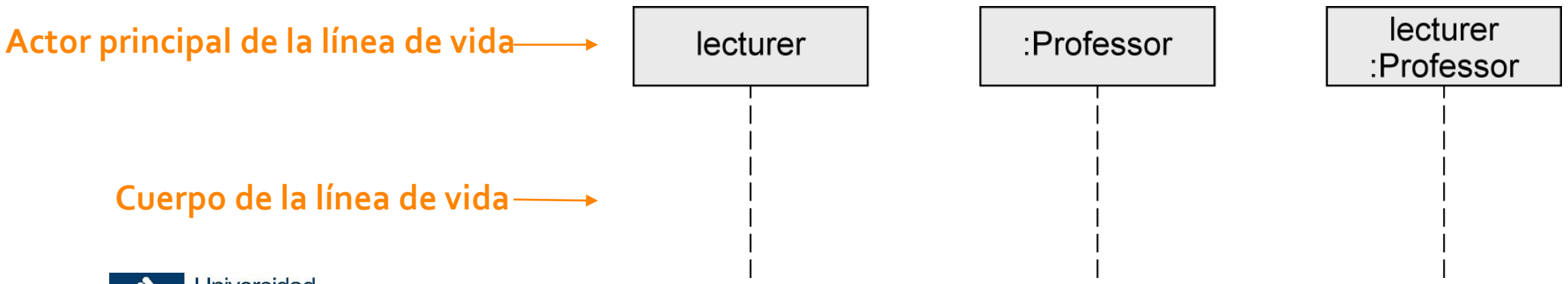


3. DIAGRAMAS DE SECUENCIA



4. ACTORES QUE INTERACTÚAN

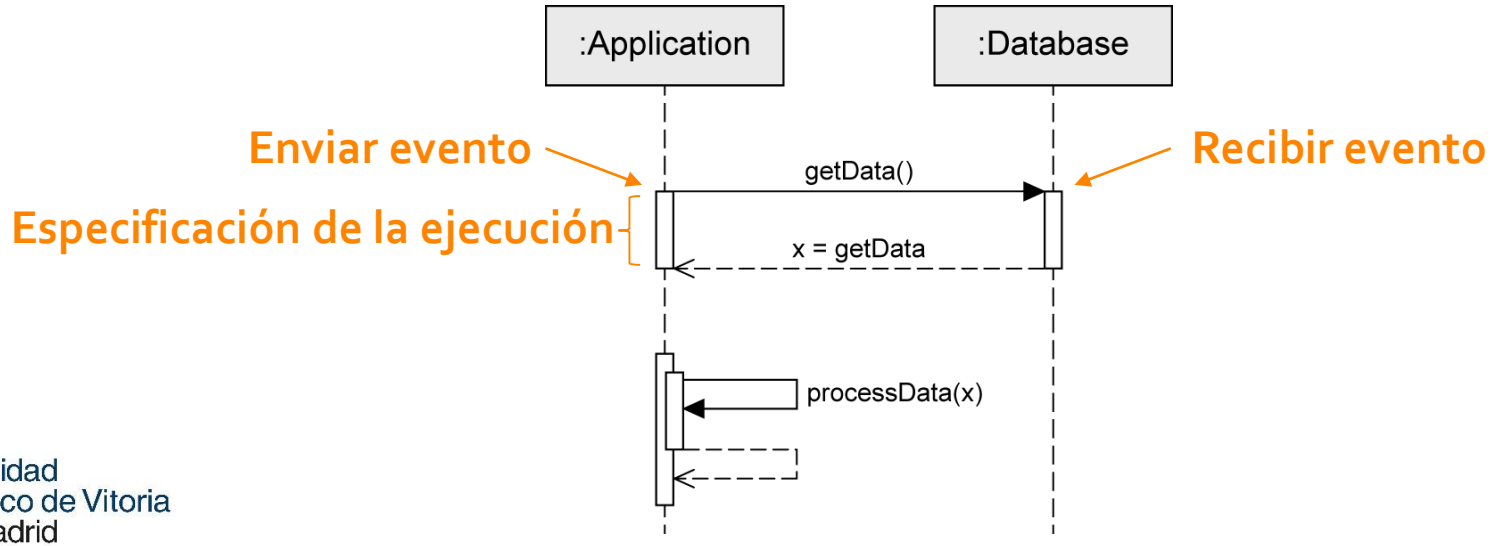
Los actores de interacción se describen como líneas de vida
Actor principal de la línea de vida.
Rectángulo que contiene la expresión rol: Class
Los roles son un concepto más general que los objetos.
El objeto puede asumir diferentes roles durante su vida.
Cuerpo de la línea de vida
Vertical, generalmente línea discontinua
Representa la vida útil del objeto asociado a él.



5. INTERCAMBIO DE MENSAJES

Interacción: secuencia de eventos
El mensaje se define mediante el envío del evento y la recepción del evento.

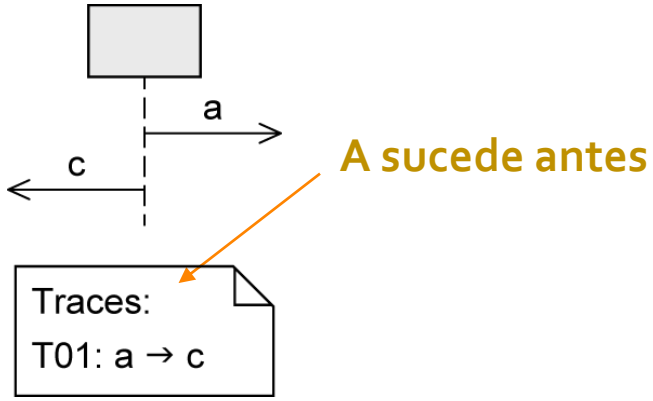
Especificación de ejecución
Barra continua
Se usa para visualizar cuando otro actor ejecuta algún comportamiento



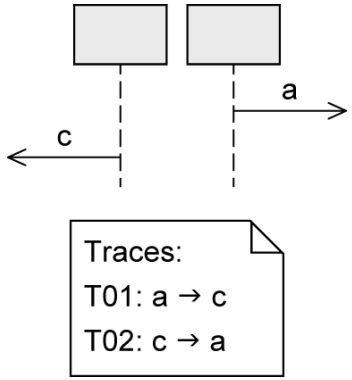
5. INTERCAMBIO DE MENSAJES

Orden de los mensajes

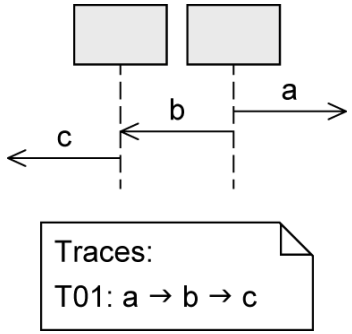
En una línea de vida



En distintas líneas de vida



… En diferentes líneas de vida intercambian mensajes



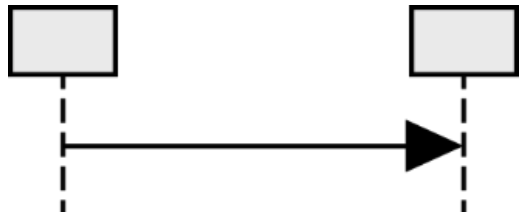
6. MENSAJES

Mensaje sincrónico

El remitente **espera** hasta recibir un mensaje de respuesta antes de continuar.

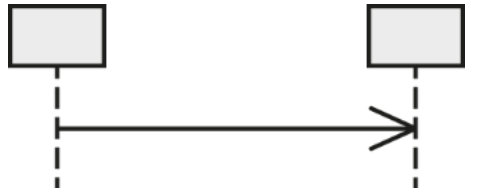
Sintaxis del mensaje: **`msg(par1,par2)`**

`msg`: nombre del mensaje.
`par`: parámetros separados por comas.



Mensaje asíncrono

El remitente continúa sin esperar un mensaje de respuesta
Sintaxis del nombre del mensaje : **`msg(par1,par2)`**



Mensaje de respuesta

Puede omitirse si el contenido y la ubicación son obvios

Sintaxis : **`att=msg(par1,par2):val`**

`att`: el valor de retorno se puede asignar opcionalmente a una variable
`msg`: thnombre del mensaje
`par`: parámetros separados por comas
`val`: valor de retorno

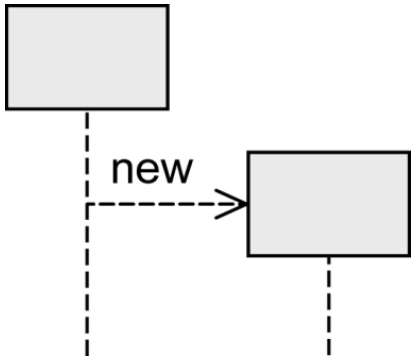
6. MENSAJES

Creación de objetos

Flecha discontínua.

La punta de flecha apunta a la cabeza de la línea de vida del objeto que se va a crear.

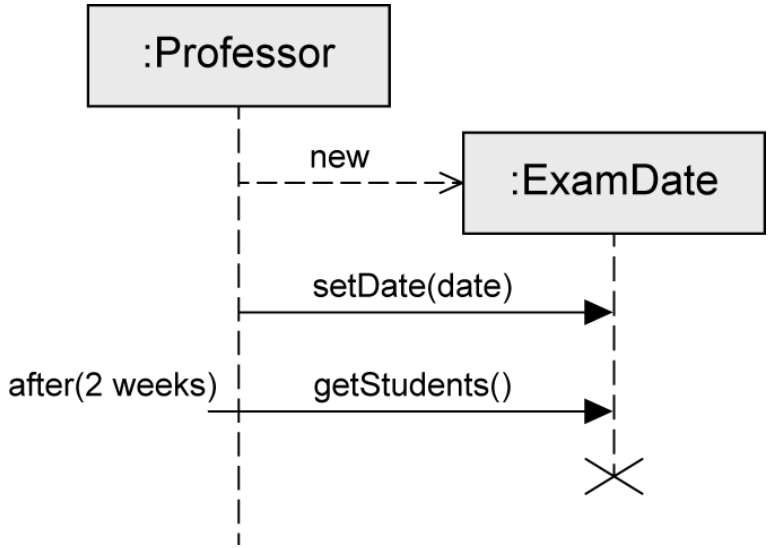
Palabra clave: New

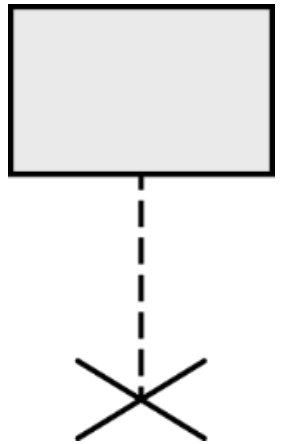


Destrucción de objetos

El objeto se elimina.

Cruz (×) al final de la línea de vida.





6. MENSAJES

Mensaje encontrado

El remitente de un mensaje es desconocido o no es relevante.

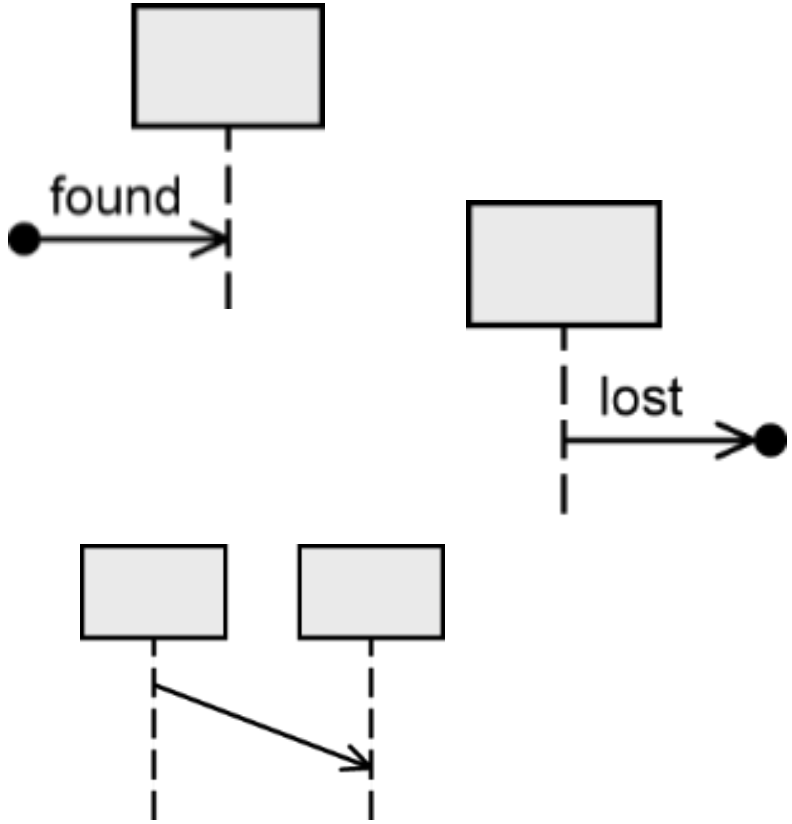
Mensaje perdido

El receptor de un mensaje es desconocido o no es relevante

Mensaje que consume mucho tiempo

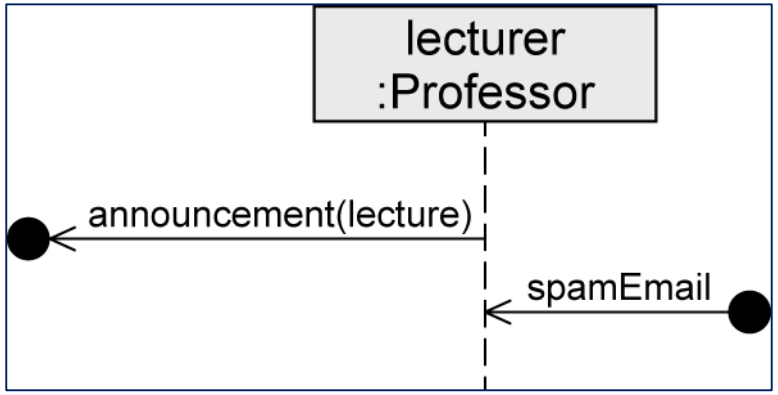
"Mensaje con duración"

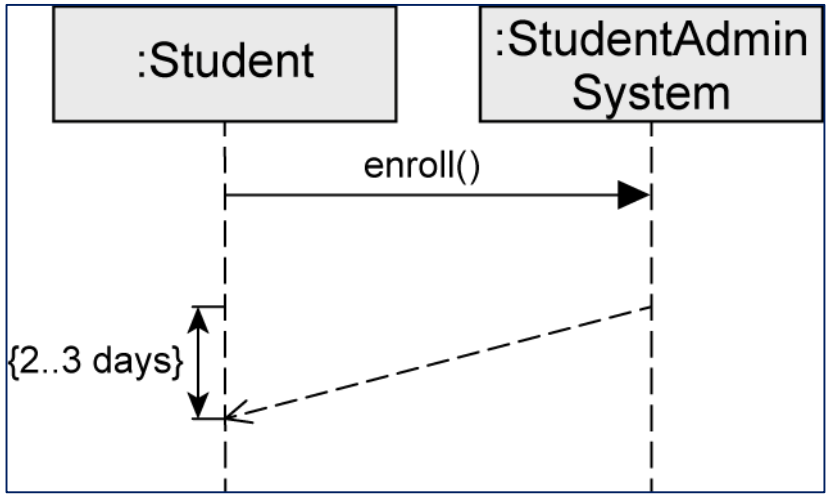
Sirve para expresar que el tiempo transcurre entre el envío y la recepción de un mensaje



Por lo general, suponemos que los mensajes se transmiten sin pérdida de tiempo.

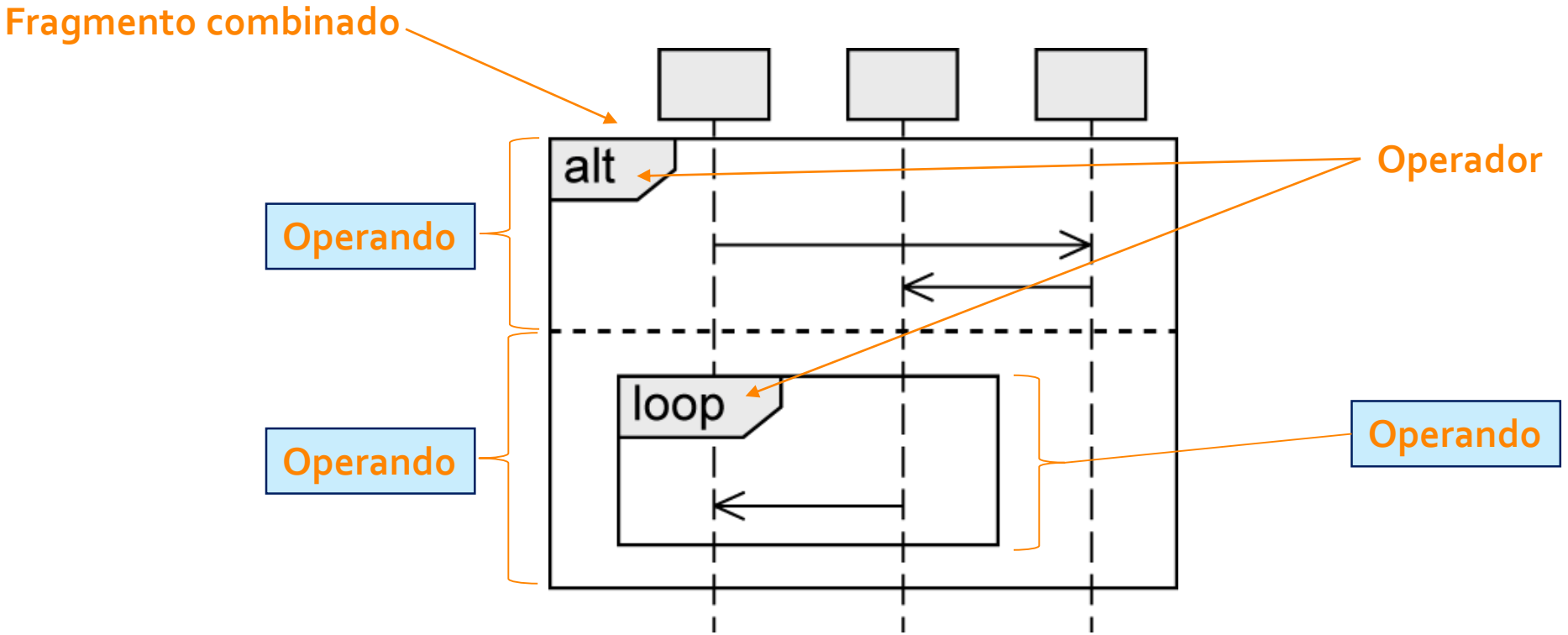
6. MENSAJES


lecturer
:Professor
announcement(lecture)
spamEmail


:Student
:StudentAdmin System
enroll()
{2..3 days}

7. FRAGMENTOS COMBINADOS

Modelar varias estructuras de control. No se espera que los diagramas de secuencia muestren lógicas de procedimientos complejos. Si fueran complejos, existen mecanismos que permiten agregar un grado de lógica en el procedimiento. Se representan como fragmentos combinados. Es una o más secuencias de procesos incluidas en un marco y ejecutadas bajo circunstancias específicas. Existen 12 tipos de operadores.



8. TIPOS DE FRAGMENTOS COMBINADOS

- El fragmento **Alternative** (denotedo "**alt**") modela estructuras if…then…else.
- El fragmento **Option** (denotado "**opt**") modela estructuras switch.
- El fragmento **Break** modela una secuencia alternativa de eventos que se procesa en lugar de todo del resto del diagrama.
- El fragmento **Parallel** (denotado "**par**") modela procesos concurrentes.
- El fragmento de secuencia **Weak** (denotado "**seq**") incluye un número de secuencias para las cuales todos los mensajes se deben procesar en un segmento anterior, antes de que el siguiente segmento pueda comenzar, pero que no impone ningún secuenciado en los mensajes que no comparten una línea de vida.
- El fragmento de secuenciado **Strict** (denotado "**strict**") incluye una serie de mensajes que se deben procesar en el orden proporcionado.
- El fragmento **Negative** (denotado "**neg**") incluye una serie de mensajes inválidos.
- El fragmento **Critical** incluye una sección crítica.
- El fragmento **Ignore** declara un mensaje o mensajes que no son de ningún interés si este aparece en el contexto actual.
- El fragmento **Consider** es el opuesto del fragmento Ignore: cualquier mensaje que no se incluya en el fragmento Consider se debería ignorar.
- El fragmento **Assertion** (denotado "**assert**") designa que cualquier secuencia es inválida si no se muestra como un operando de la aserción. El fragmento **Loop** incluye una serie de mensajes que están repetidos.

8. TIPOS DE FRAGMENTOS COMBINADOS

	Operator	Purpose
Ramas y bucles	`alt`	Interacción alternativa
	`opt`	Intreacción opcional
	`loop`	Interacción repetida
	`break`	Interacción de excepción
Concurrencia y otros	`seq`	Weak
	`strict`	Orden estricto
	`par`	Interacción concurrente
	`critical`	Interacción crítica
Filtros y aserciones	`ignore`	Interacción irrelevante
	`consider`	Interacción relevante
	`assert`	Asserted interaction
	`neg`	Invalid interaction

Universidad Francisco de Vitoria UFV Madrid

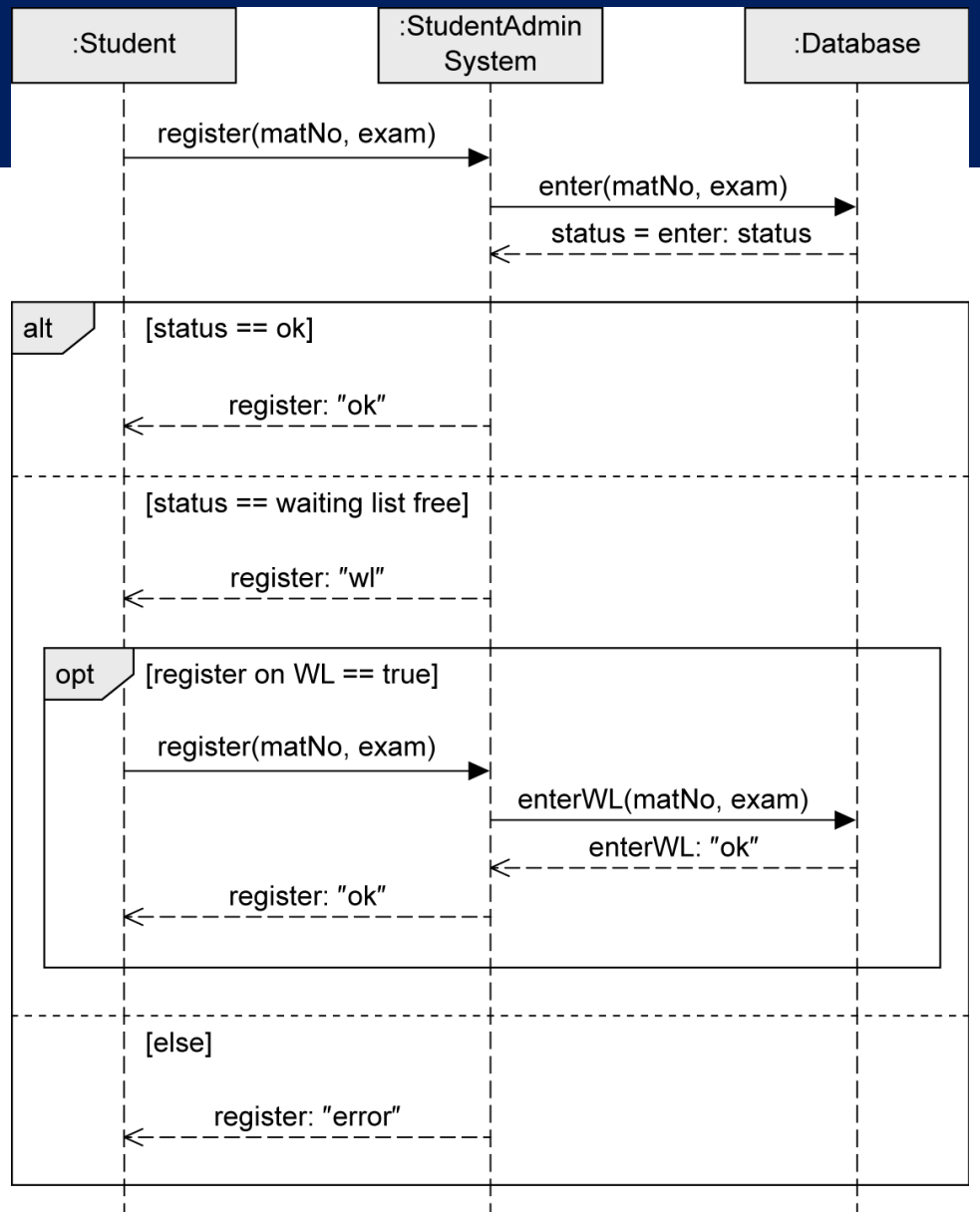
ALT

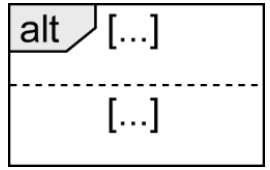
Para modelar secuencias alternativas.
Similar a la declaración **switch** en Java
Las guardas se utilizan para seleccionar la única ruta a ejecutar.
Guardas

- Se diseña entre corchetes
- default: `true`
- predefined: `[else]`

Varios operandos

Guardas deben estar separadas para asegurar que no se produce un comportamiento inesperado





Universidad Francisco de Vitoria UFV Madrid

OPT

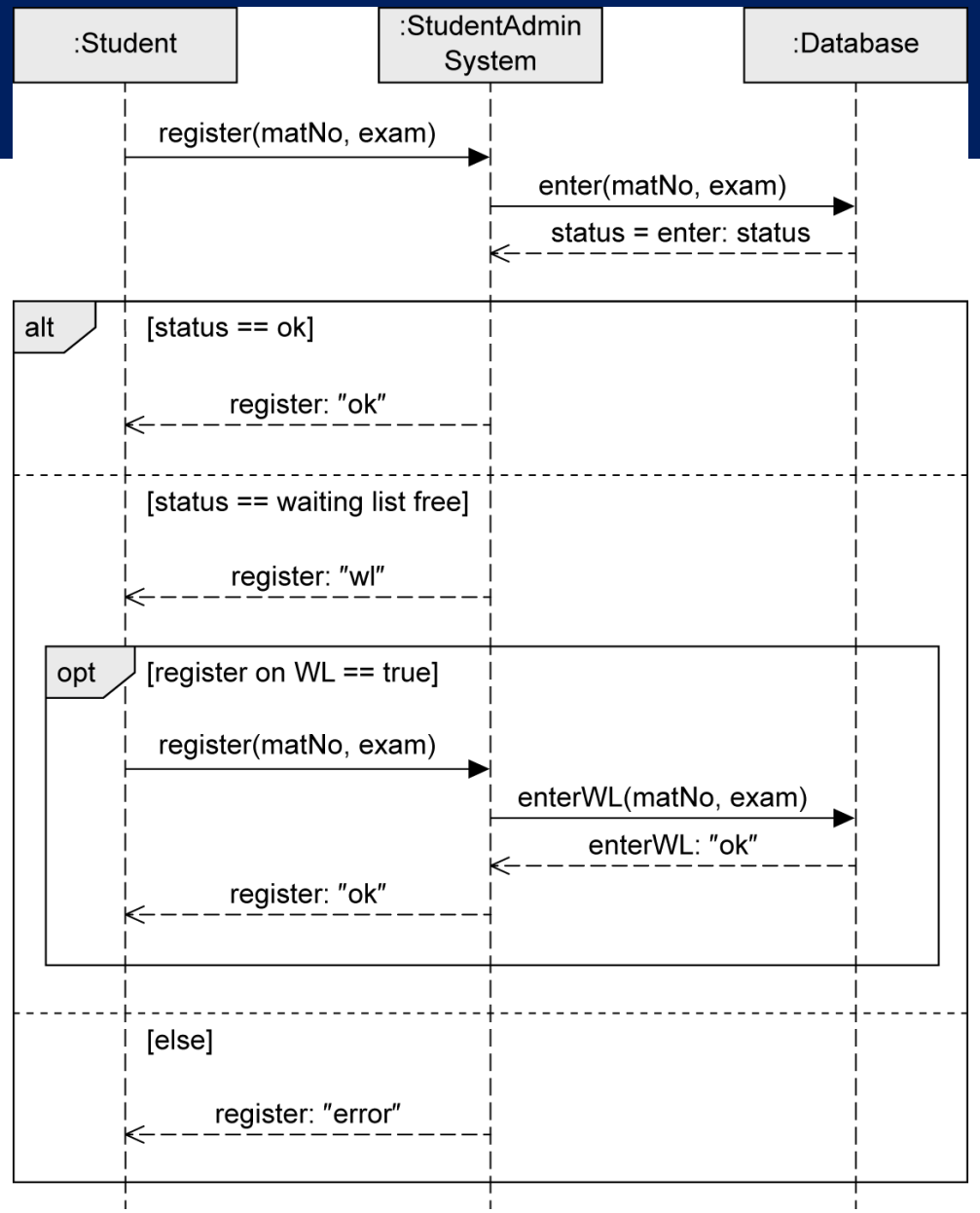
Para modelar una secuencia opcional

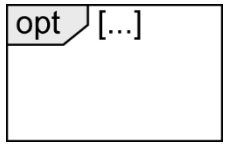
La ejecución real en tiempo de ejecución depende del guardia

Exactamente un operando

Similar a **if sin else**.

equivalente a un **fragmento alt con dos operandos**, uno de los cuales se encuentra vacío.

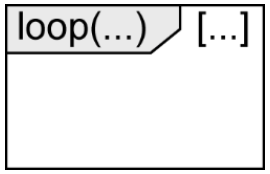


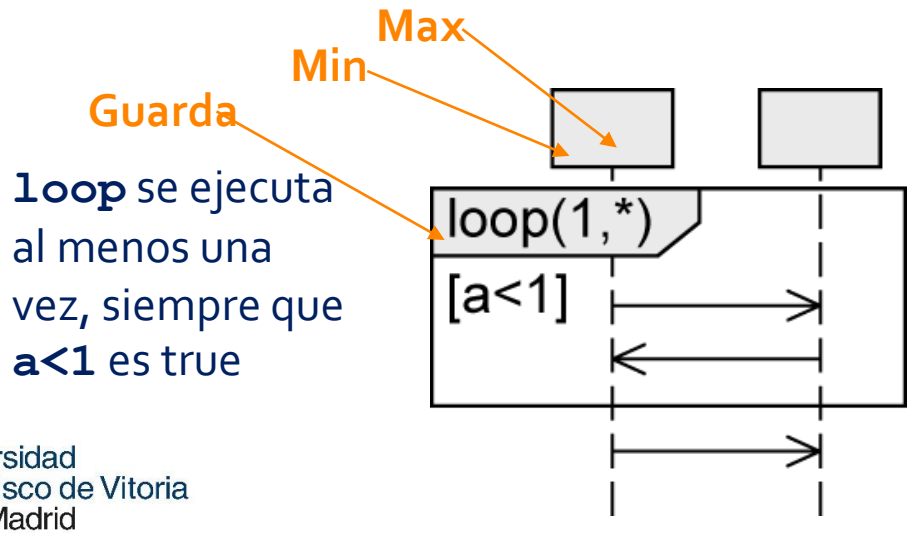


LOOP

Sirve para expresar que una secuencia debe ejecutarse **repetidamente**

Bucle de palabras clave seguido del número mínimo / máximo de iteraciones (mín. Máx.) O (mín., Máx.)
predeterminado: (*) .. sin límite superior
Guardia
Evaluado tan pronto como haya tenido lugar el número mínimo de iteraciones
Verificado para cada iteración dentro de los límites (mínimo, máximo)
Si la guardia se evalúa como falsa, la ejecución del bucle finaliza.





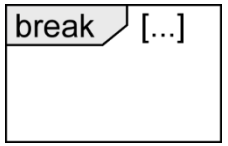
`loop` se ejecuta al menos una vez, siempre que **`a<1`** es true

Notation alternativas:

```
loop(3,8) = loop(3..8)
loop(8,8) = loop (8)
loop = loop (*) = loop(0,*)
```

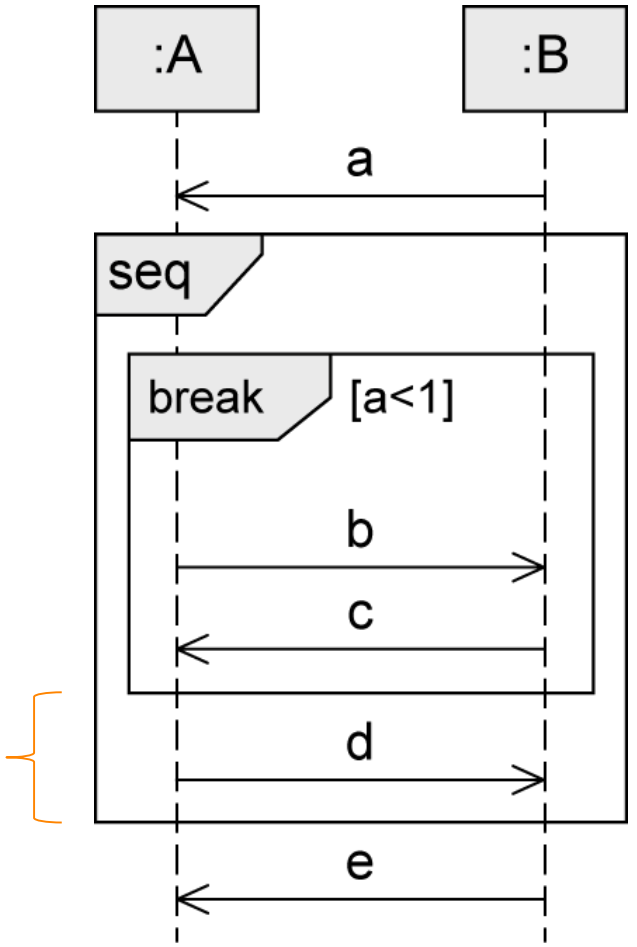
BREAK

Forma simple de manejo de excepciones



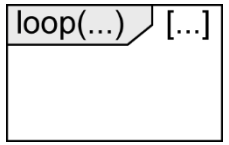
Si la guarda es verdadera:

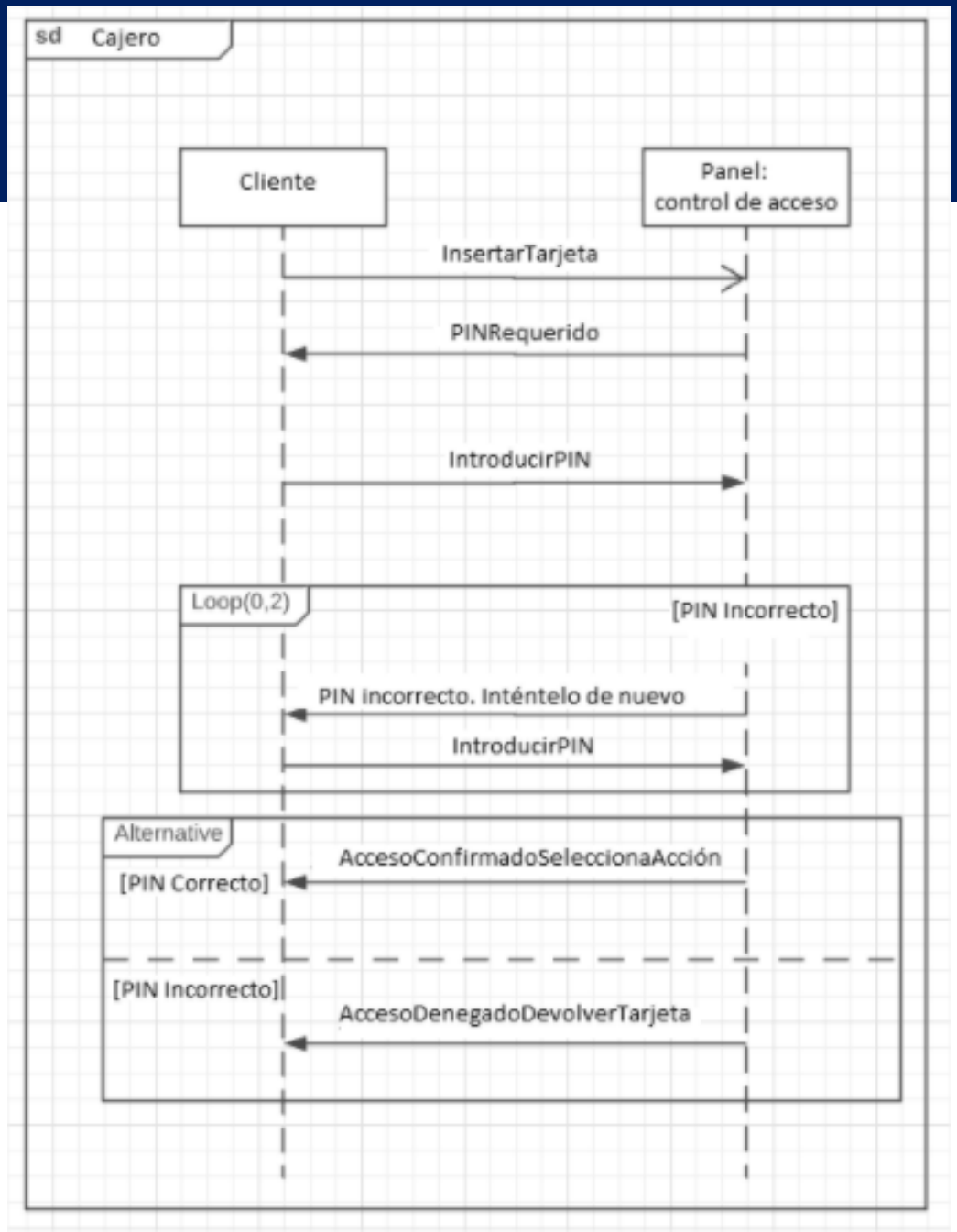
- Las interacciones dentro de este operando se ejecutan.
- Se omiten las operaciones restantes del fragmento.
- La interacción continúa en el siguiente fragmento de nivel superior
- Comportamiento diferente al fragmento opt.



No se ejecuta si BREAK se ejecuta

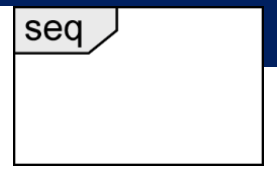
LOOP, BREAK - EJEMPLO





SEQ

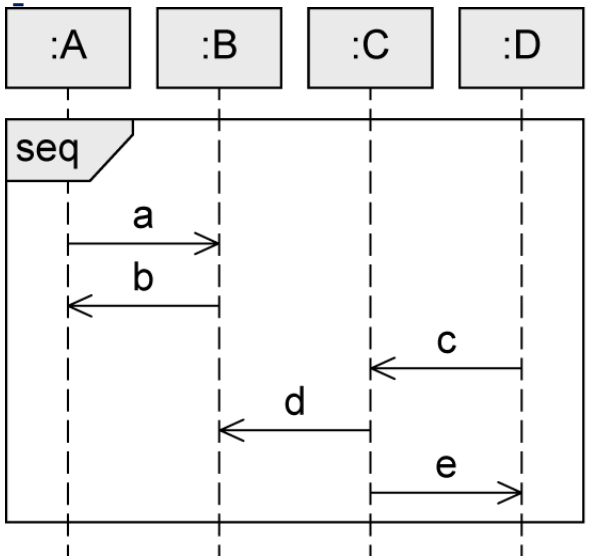
Orden predeterminado de eventos. Secuenciación débil:



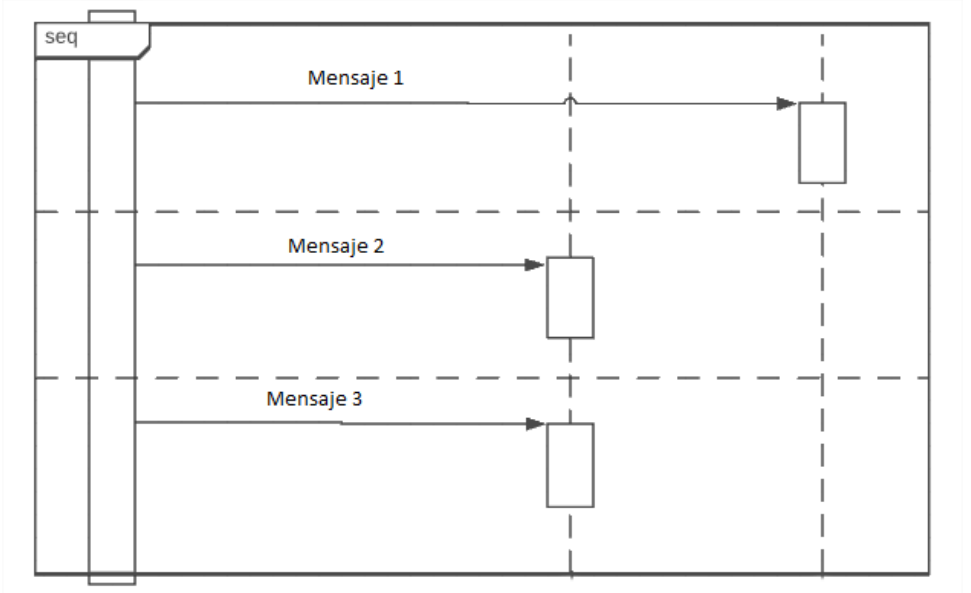
El orden de los eventos dentro de cada uno de los operandos se mantiene en el resultado.

Los eventos en diferentes líneas de vida de diferentes operandos pueden aparecer en cualquier orden.

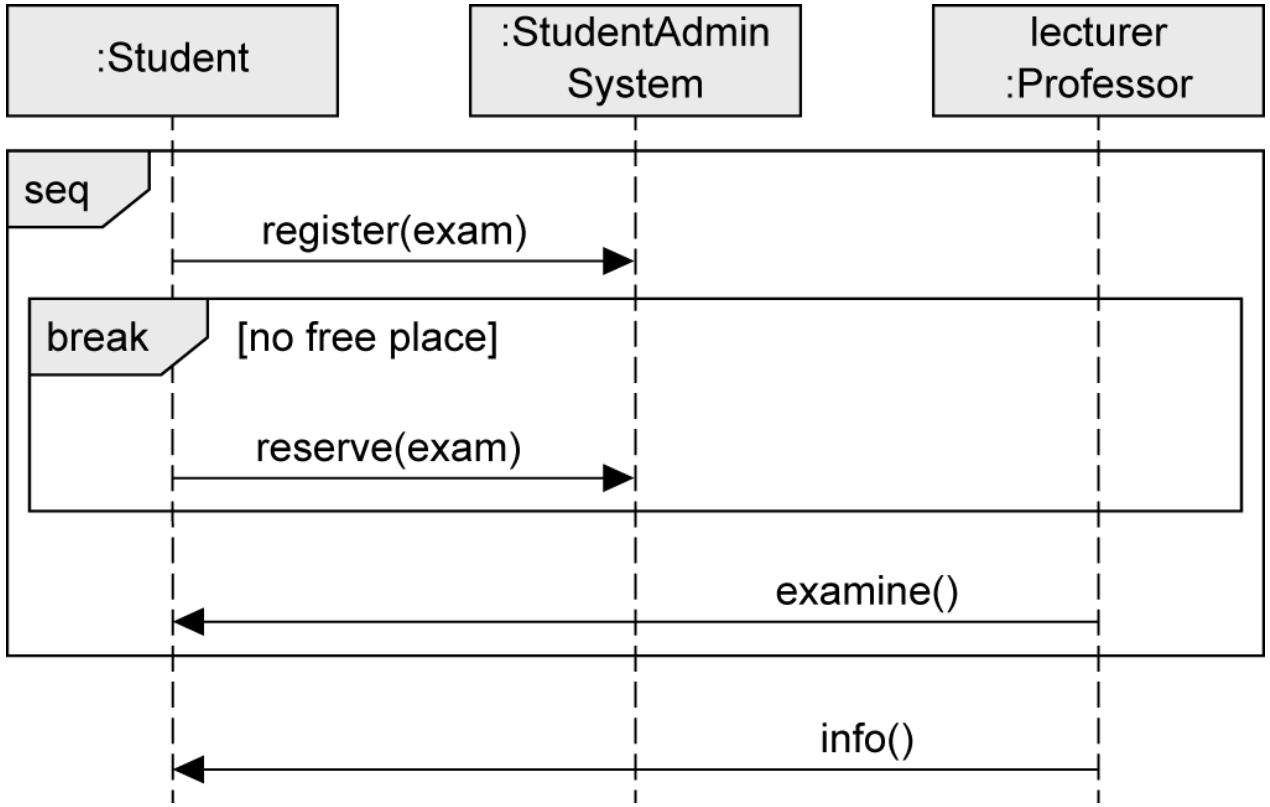
Los eventos en la misma línea de vida de diferentes operandos se ordenan de tal manera que un evento del primer operando viene antes que el del segundo operando..



Traces:
T01: a → b → c → d → e
T02: a → c → b → d → e
T03: c → a → b → d → e



SEQ - EJEMPLO

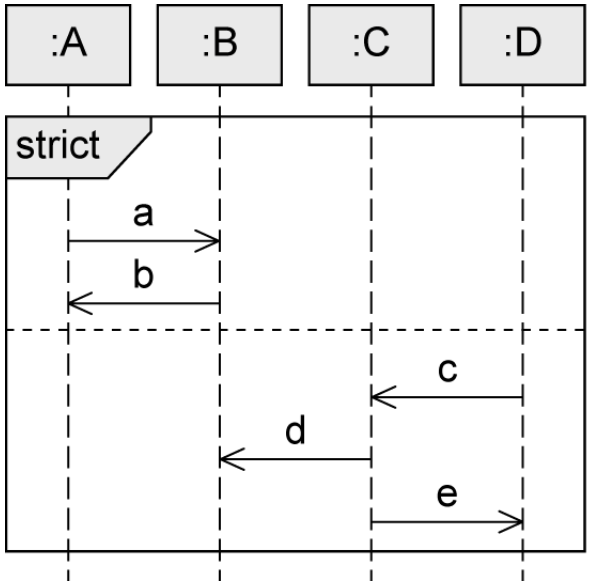

:Student
:StudentAdmin System
lecturer :Professor
seq
register(exam)
break
[no free place]
reserve(exam)
examine()
info()

STRICT

Interacción secuencial con orden

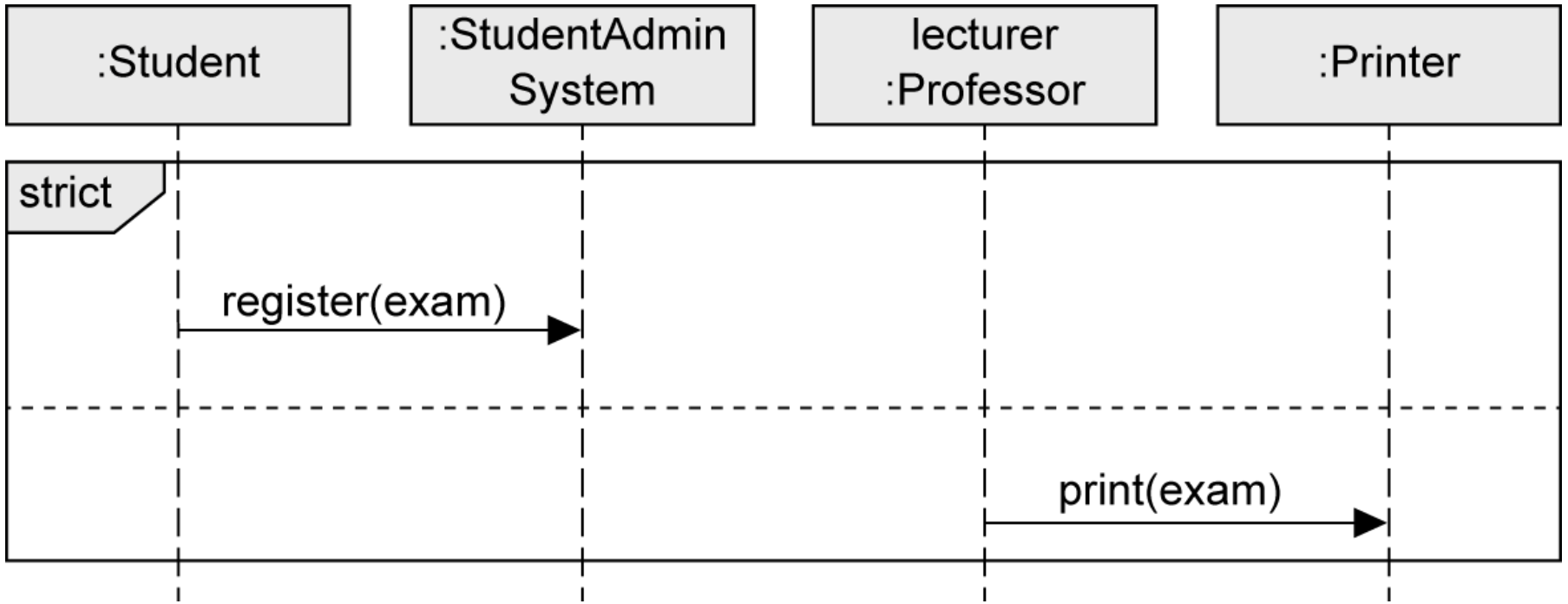
El orden de ocurrencia de eventos en diferentes líneas de vida entre diferentes operandos es significativo

Los mensajes en un operando que está más arriba en el eje vertical siempre se intercambian antes que los mensajes en un operando que está más abajo en el eje vertical



Traces:
T01: a → b → c → d → e

STRICT - EJEMPLO



PAR

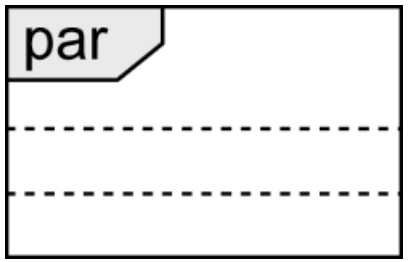
Se utiliza para dejar de lado el orden cronológico entre mensajes en diferentes operandos

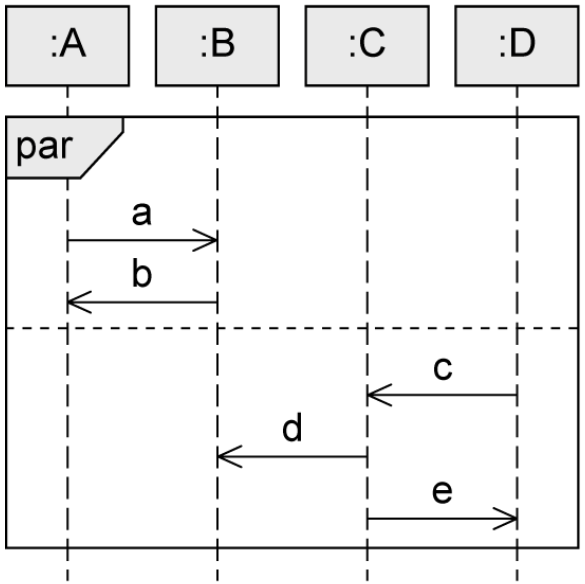
Las rutas de ejecución de diferentes operandos se pueden intercalar

Se deben respetar las restricciones de cada operando

El orden de los diferentes operandos es irrelevante

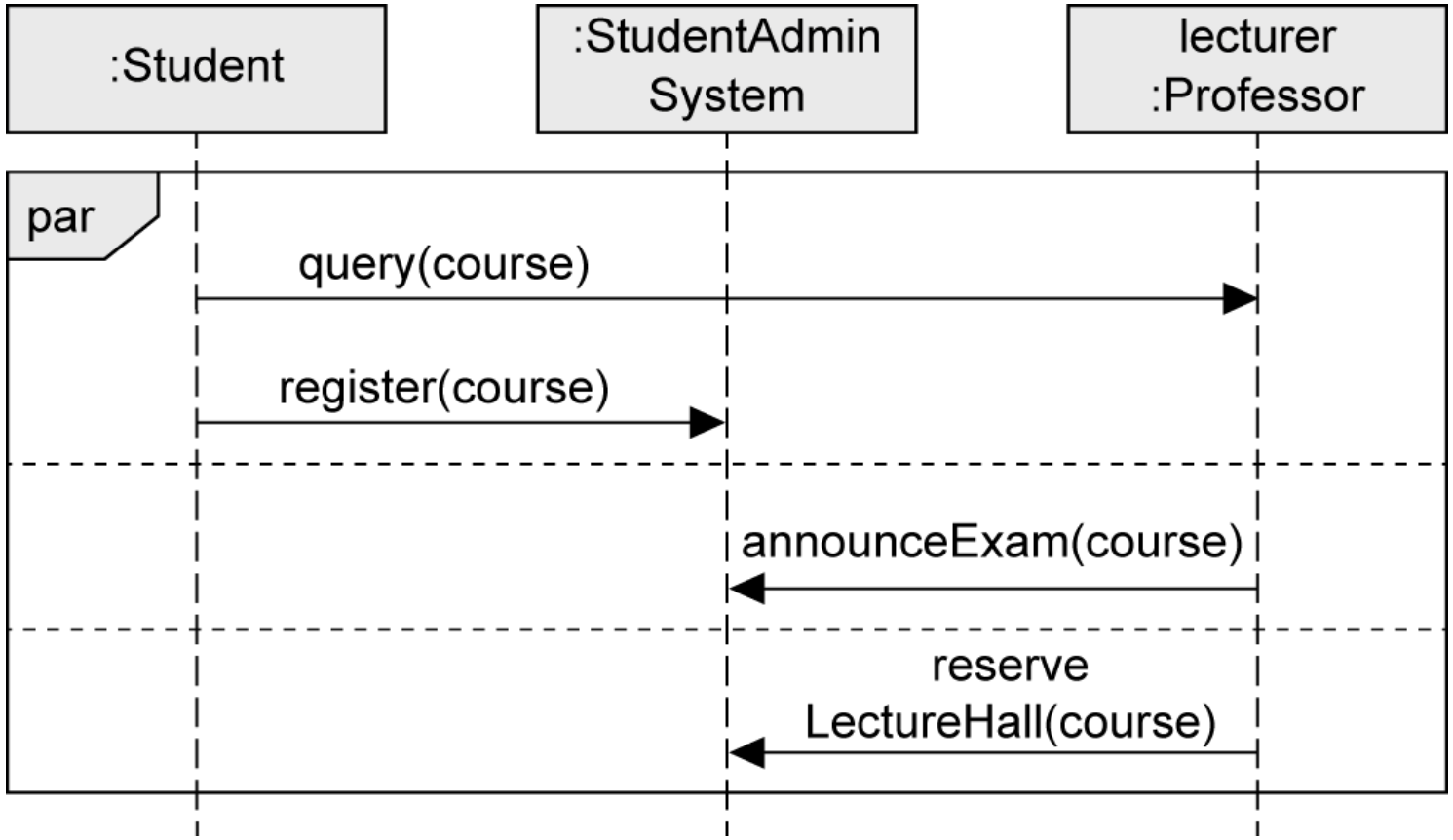
Se produce simultaneidad. Sin auténtico paralelismo.





Traces:
T01: a → b → c → d → e
T02: a → c → b → d → e
T03: a → c → d → b → e
T04: a → c → d → e → b
T05: c → a → b → d → e
T06: c → a → d → b → e
T07: c → a → d → e → b
T08: c → d → a → b → e
T09: c → d → a → e → b
T10: c → d → e → a → b

PAR - EJEMPLO

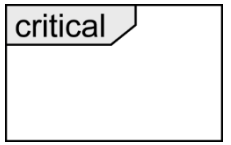

:Student
:StudentAdmin System
lecturer :Professor
par
query(course)
register(course)
announceExam(course)
reserve LectureHall(course)

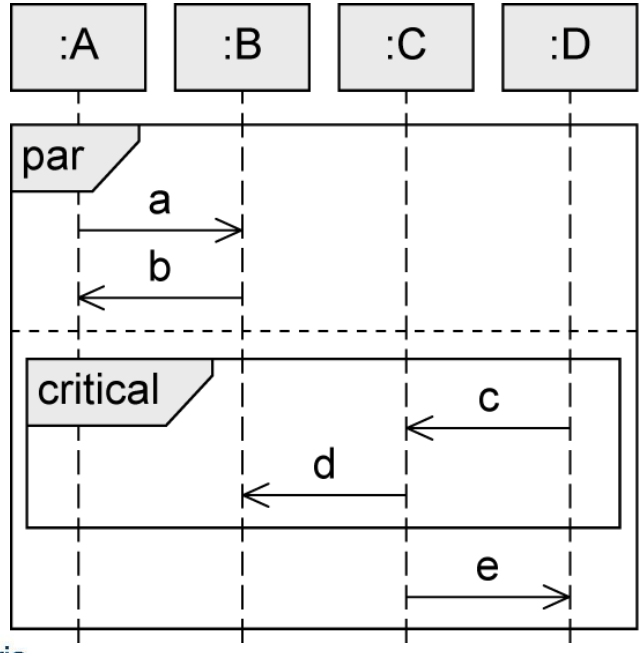
CRITICAL

Área crítica en la interacción (un operando)

Para asegurarse de que ciertas partes de una interacción no sean interrumpidas por eventos inesperados

Orden dentro de crítica: secuencia de orden predeterminada

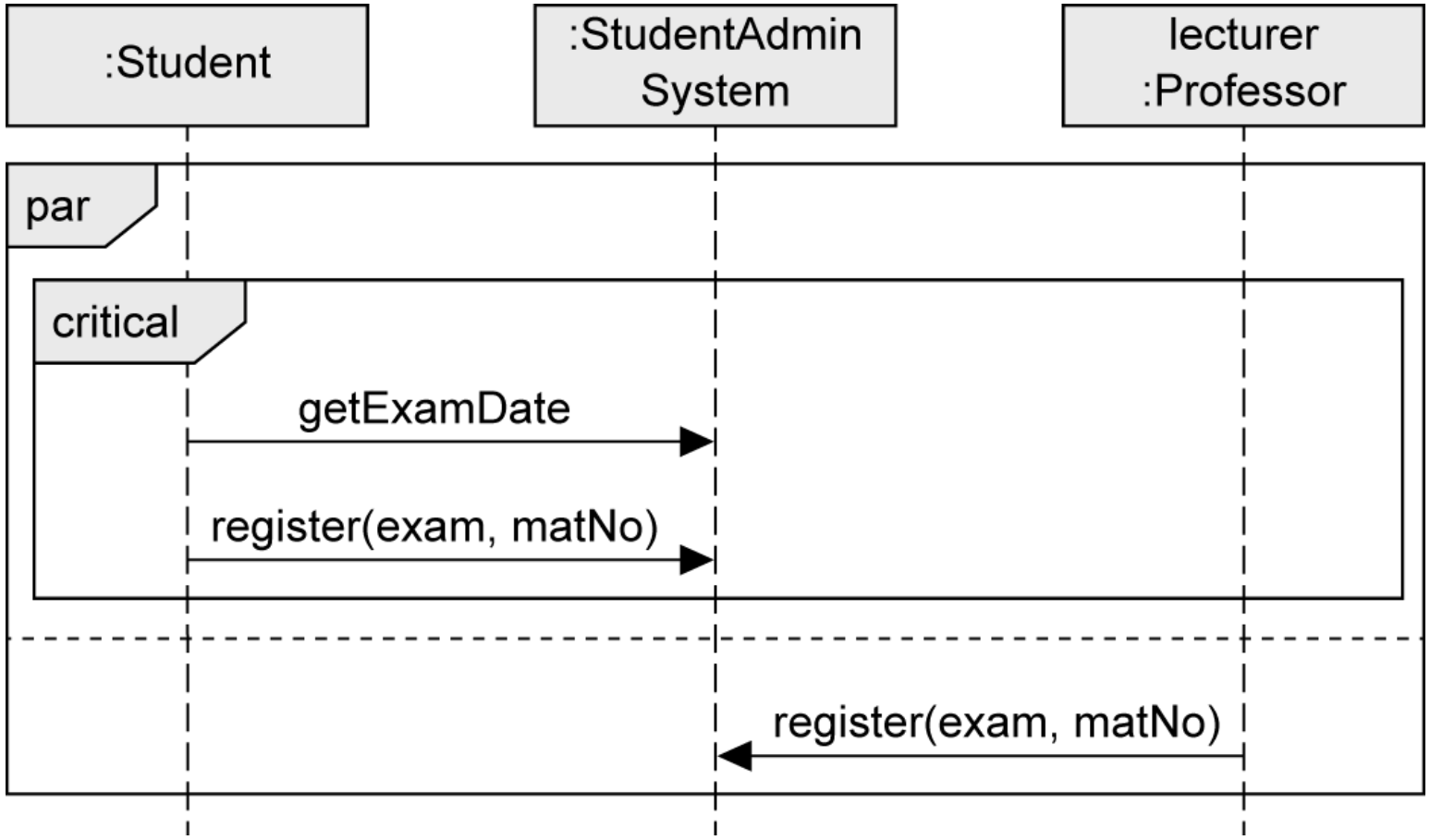




Traces:
T01: a → b → c → d → e
T02: a → c → d → b → e
T03: a → c → d → e → b
T04: c → d → a → b → e
T05: c → d → a → e → b
T06: c → d → e → a → b

Universidad Francisco de Vitoria UFV Madrid

CRITICAL - EXAMPLE

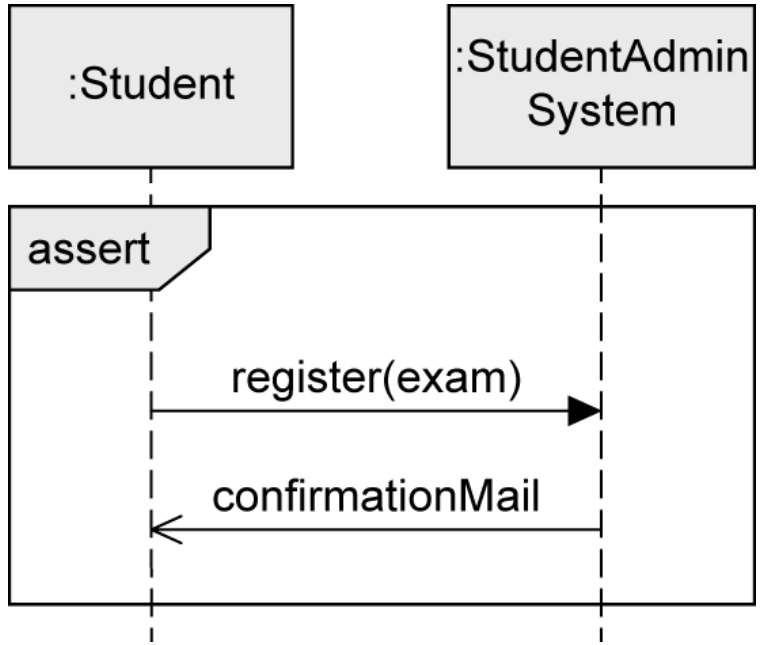


ASSERT

Para identificar **ciertas trazas diseñadas como obligatorias**

No se permiten las desviaciones que ocurren en la realidad pero que no están incluidas en el diagrama.

Exactamente un operando



NEG

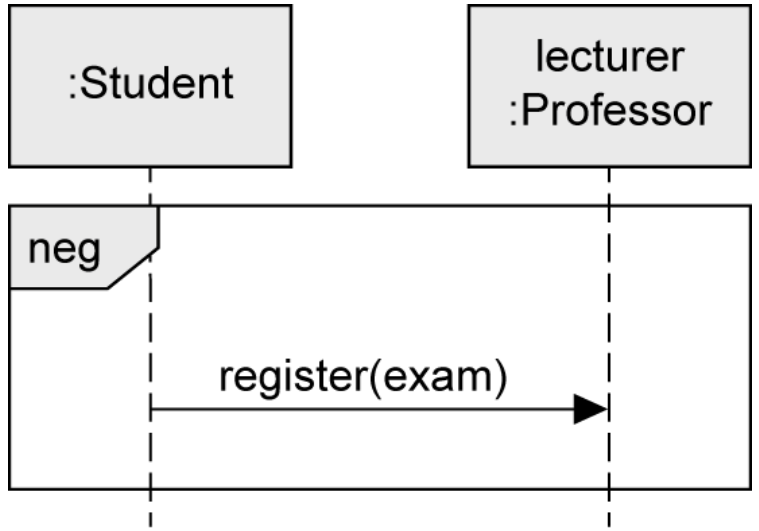
Para diseñar interacciones no válidas

Describir situaciones que no deben ocurrir

Exactamente un operando

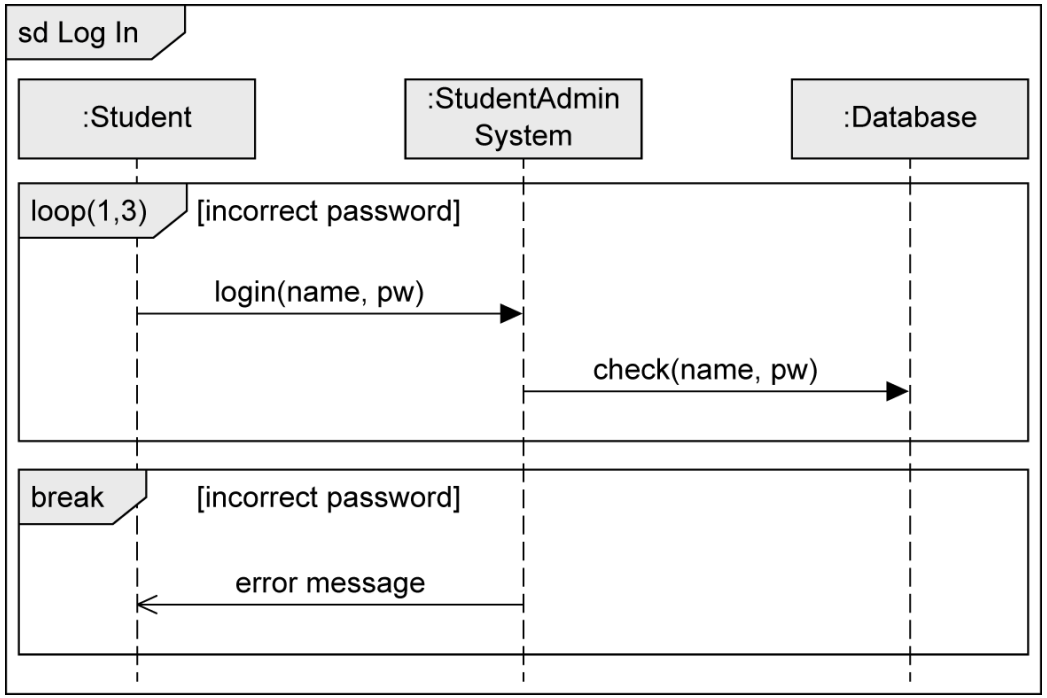
Propósito

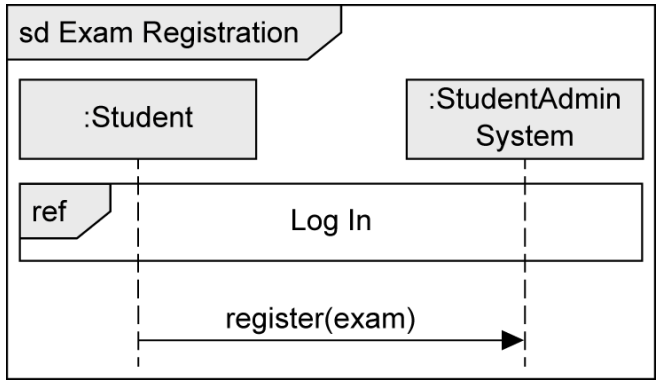
- Resaltar explícitamente los errores que ocurren con frecuencia
- Representar secuencias relevantes e incorrectas



REFERENCIA DE INTERACCIÓN

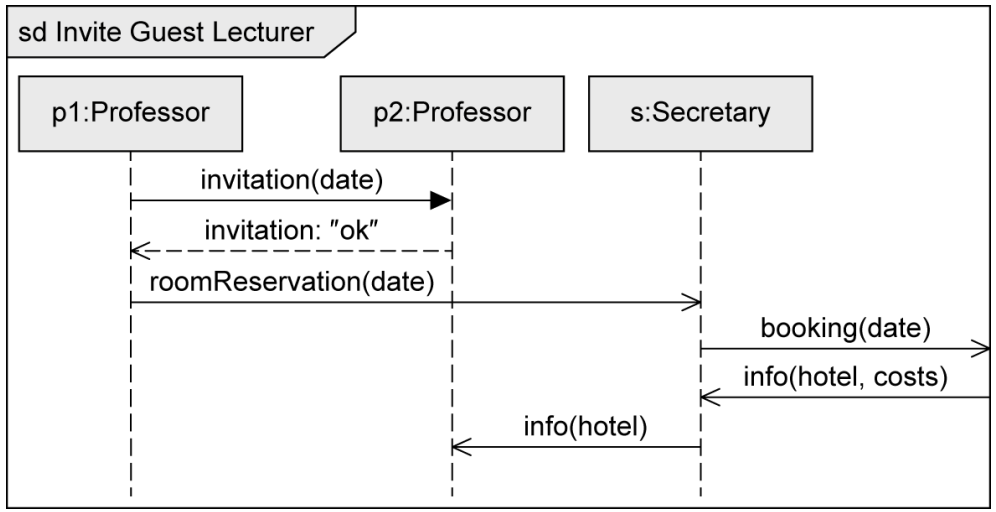
Integra un diagrama de secuencia en otro diagrama de secuencia

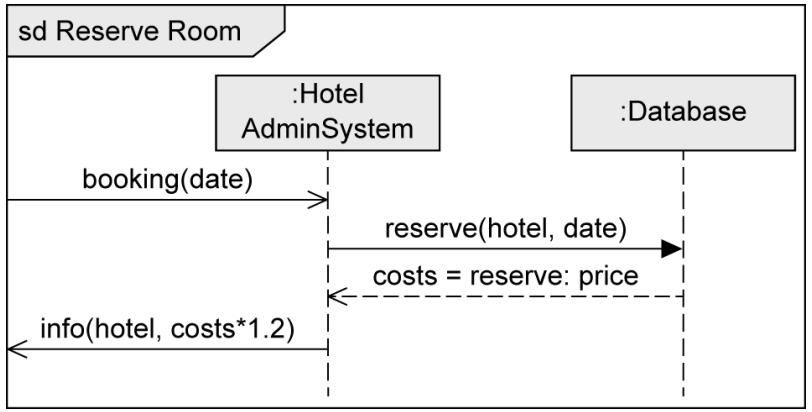




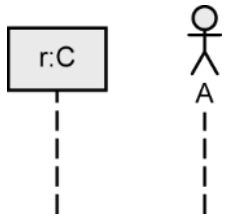
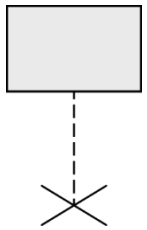
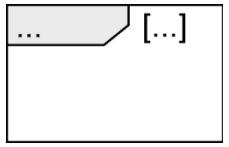
GATE

Le permite enviar y recibir mensajes más allá de los límites del fragmento de interacción.

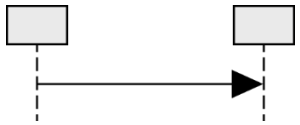
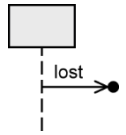
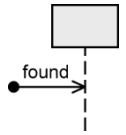




ELEMENTOS DE NOTACIÓN

Name	Notation	Description
Lifeline		Interaction partners involved in the communication
Destruction event		Time at which an interaction partner ceases to exist
Combined fragment		Control constructs

NOTATION ELEMENTS (2/2)

Name	Notation	Description
Synchronous message		Sender waits for a response message
Response message		Response to a synchronous message
Asynchronous communication		Sender continues its own work after sending the asynchronous message
Lost message		Message to an unknown receiver
Found message		Message from an unknown sender