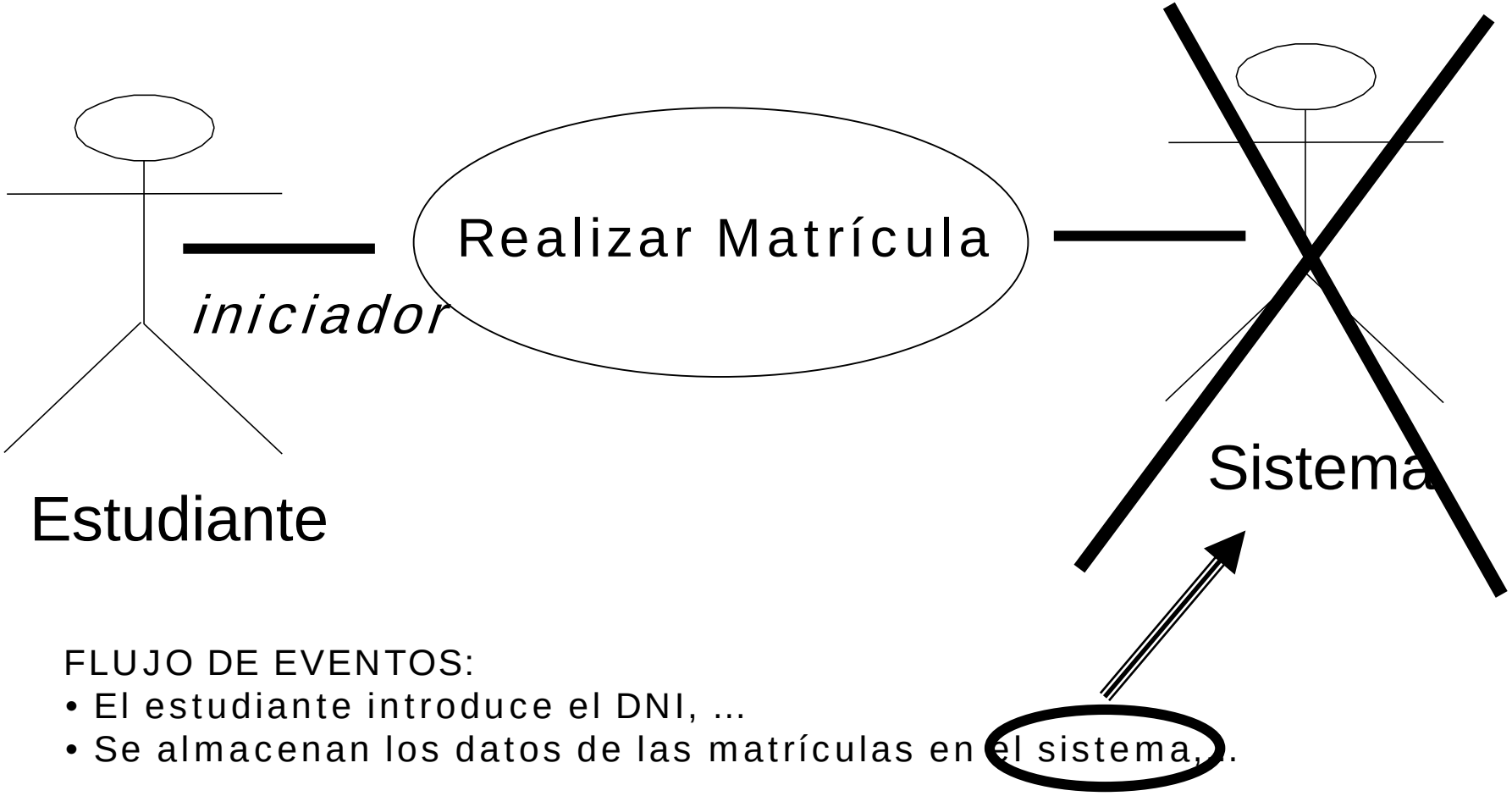
Errores típicos en captura de requisitos

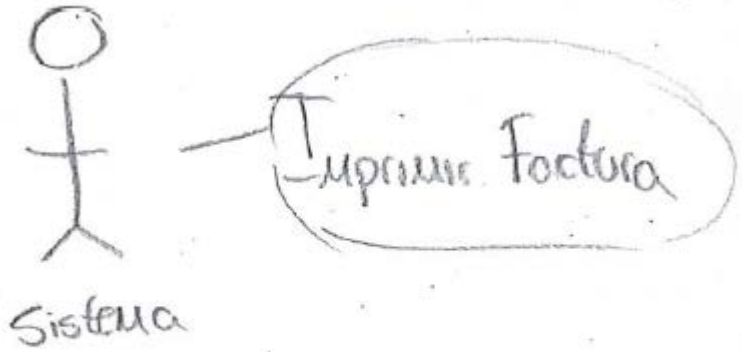
1) Error típico: poner el sistema como actor del CU

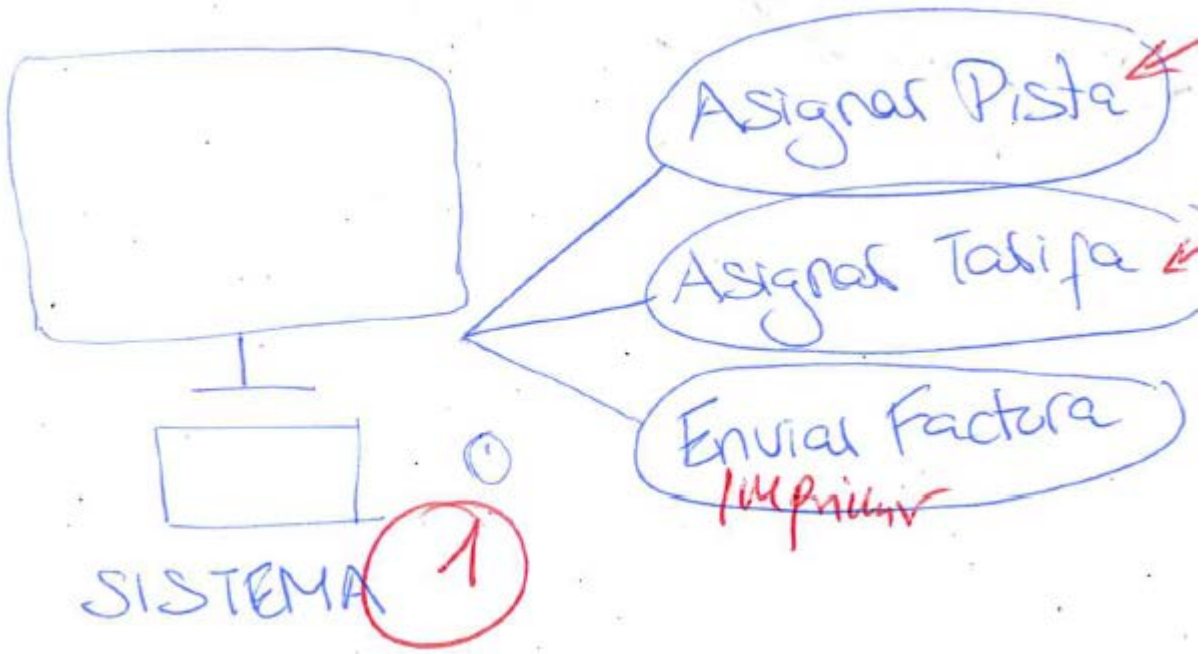


FLUJO DE EVENTOS:

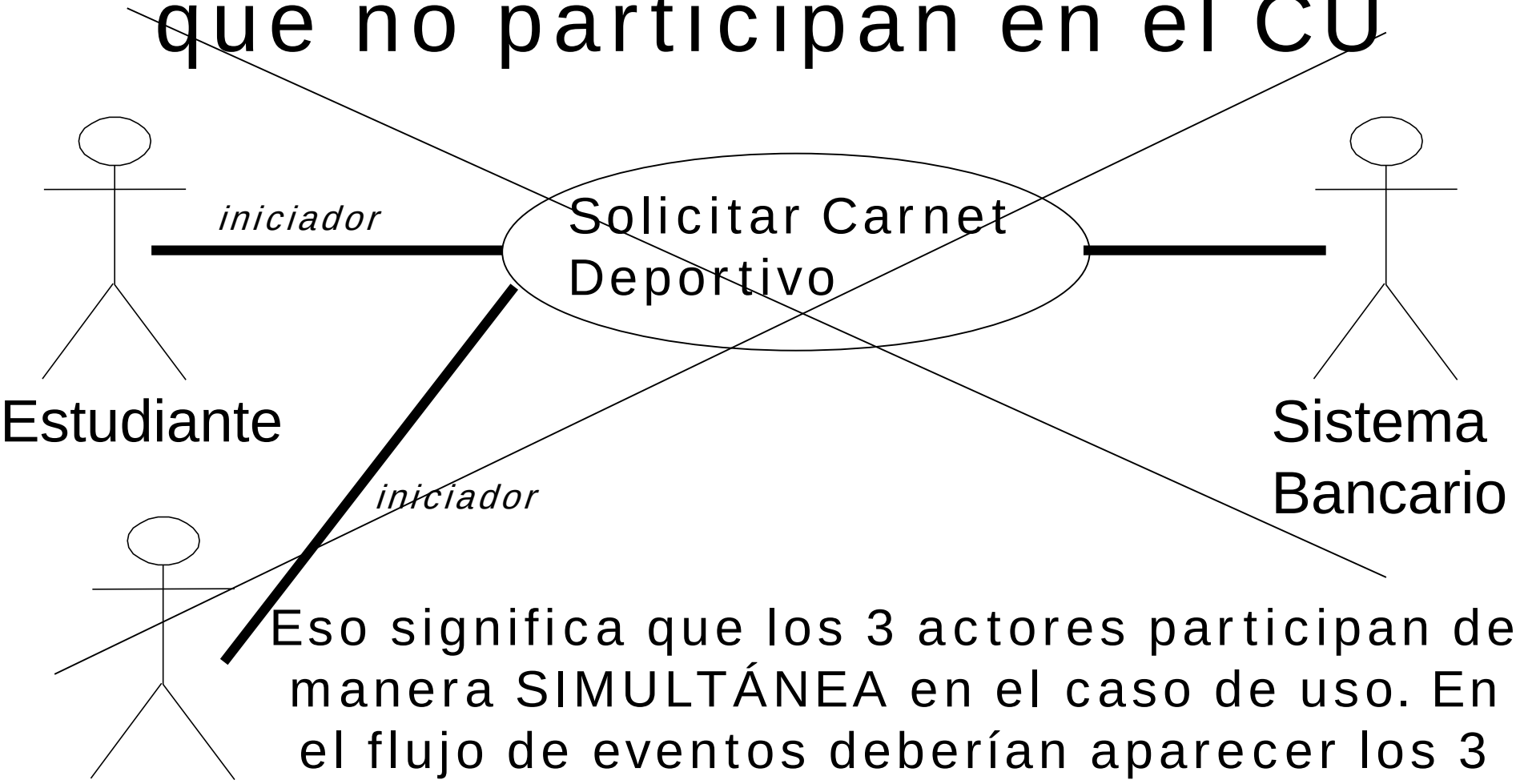
- El estudiante introduce el DNI, ...
- Se almacenan los datos de las matrículas en el sistema, ...

NO SE TRATA DE UN SISTEMA EXTERNO
SINO DEL PROPIO SISTEMA (EL C.U. ES PARTE DE ÉL)


Imprimir Factura
Sistema
1

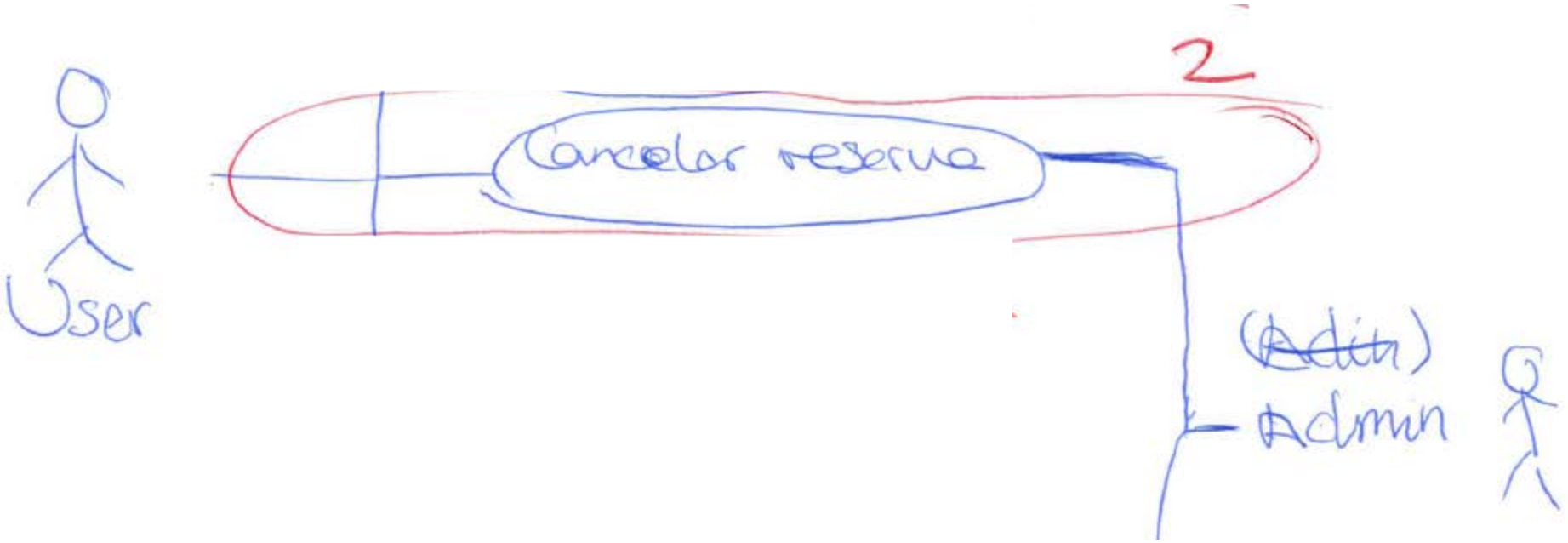

Asignar Pista
Asignar Tarifa
Enviar Factura
IMPRIMIR
SISTEMA
1

2) Error típico: poner actores que no participan en el CU

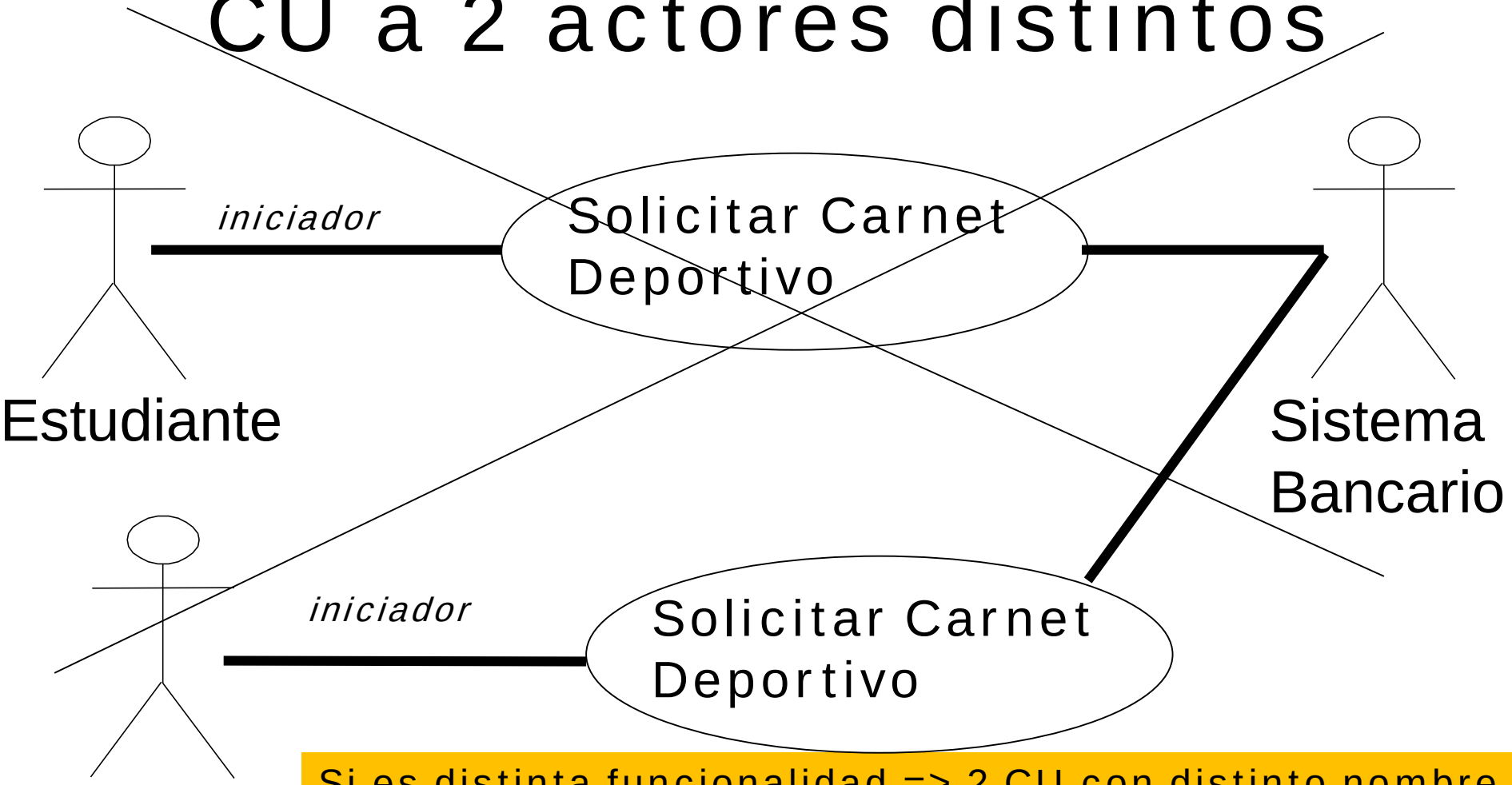


Eso significa que los 3 actores participan de manera SIMULTÁNEA en el caso de uso. En el flujo de eventos deberían aparecer los 3 actores proporcionando o recibiendo datos.

Seguramente se quería expresar que tanto estudiantes como profesores pueden solicitar el carnet deportivo

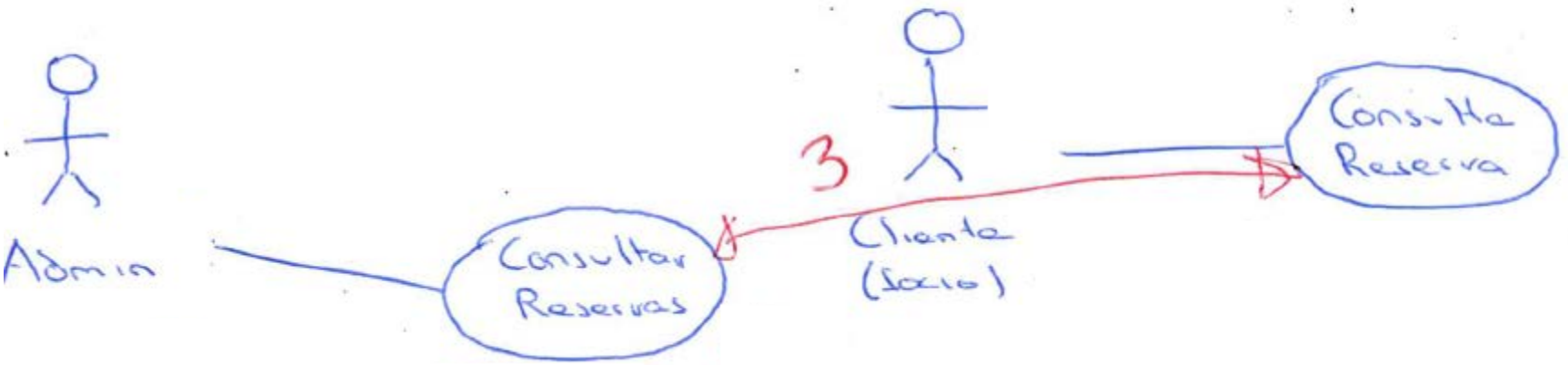

2
Cancelar reserva
User
(~~Edit~~)
Admin

3) Error típico: poner el mismo CU a 2 actores distintos

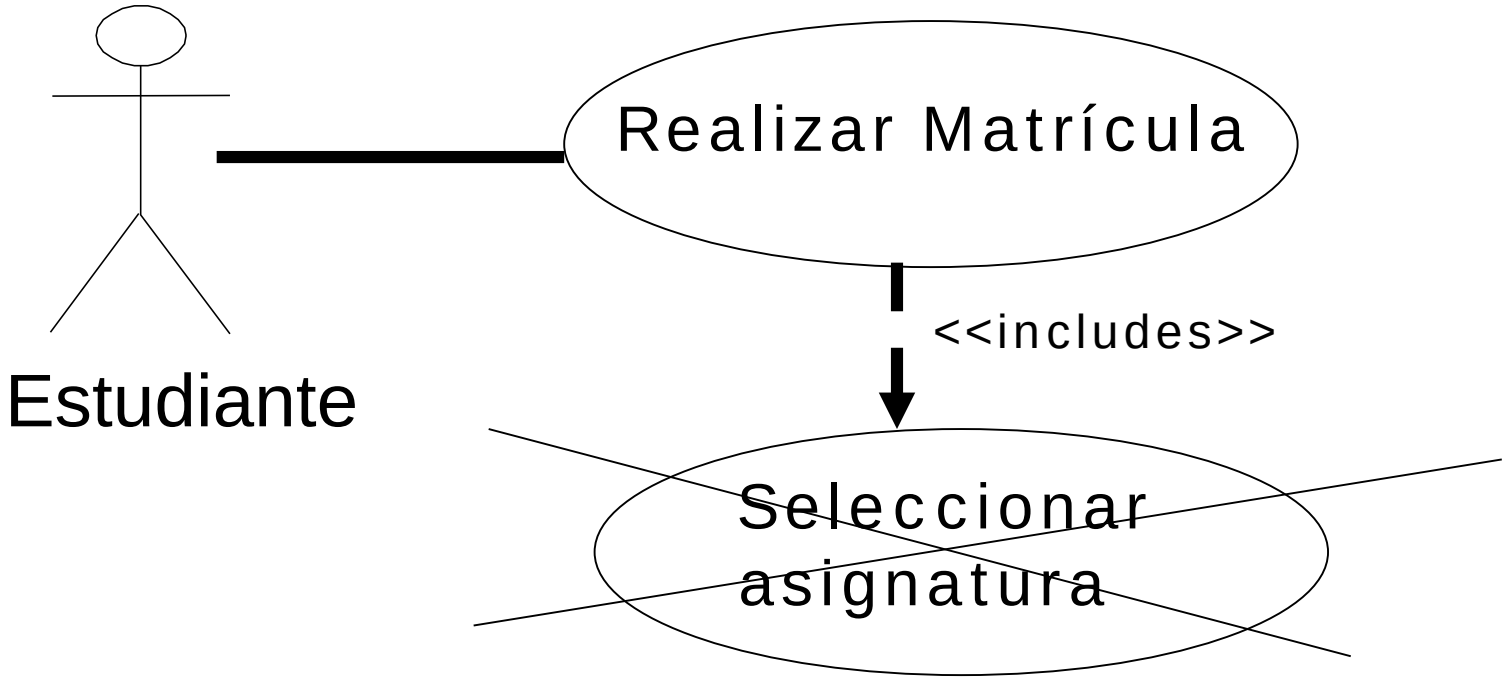


Si es distinta funcionalidad => 2 CU con distinto nombre

Si es la misma funcionalidad => un único CU asociado a un único ACTOR (por ejemplo: Universitario, o Persona...)


Admin
Consultar Reservas
3
Cliente (Socio)
Consulta Reserva

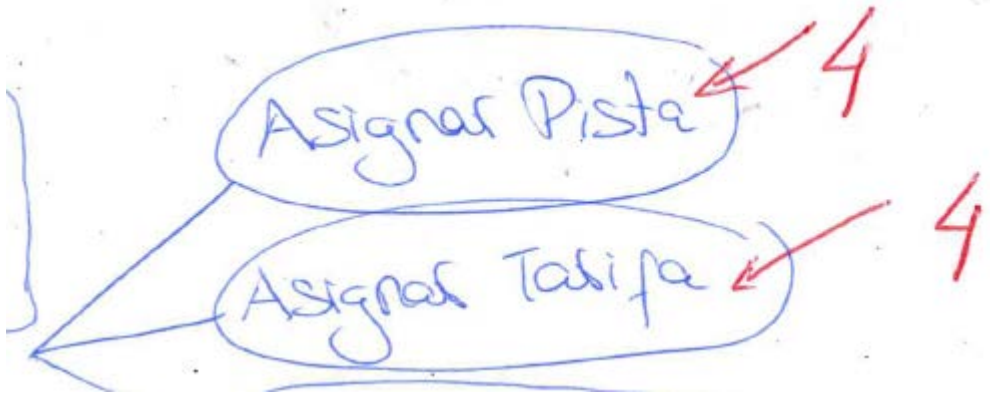
4) Errores típicos en CU: definir funcionalidad interna como CU



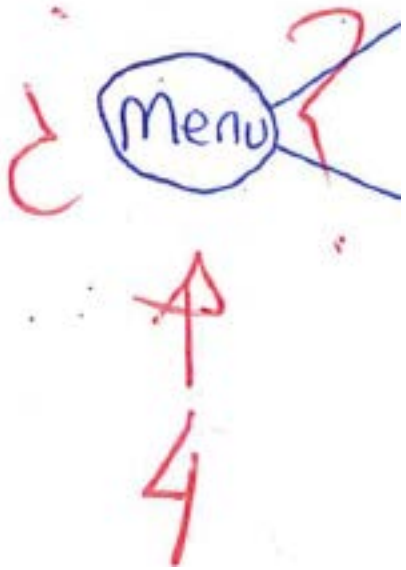
Si al realizar la matrícula, se seleccionan en la interfaz de usuario las asignaturas en las que matricularse NO es CU

F.Eventos de “Realizar Matrícula”
-El sistema muestra las asignaturas disponibles
-El estudiante selecciona las asignaturas
-El sistema registra la matrícula en ellas ...

Funcionalidad interna en Flujo de Eventos

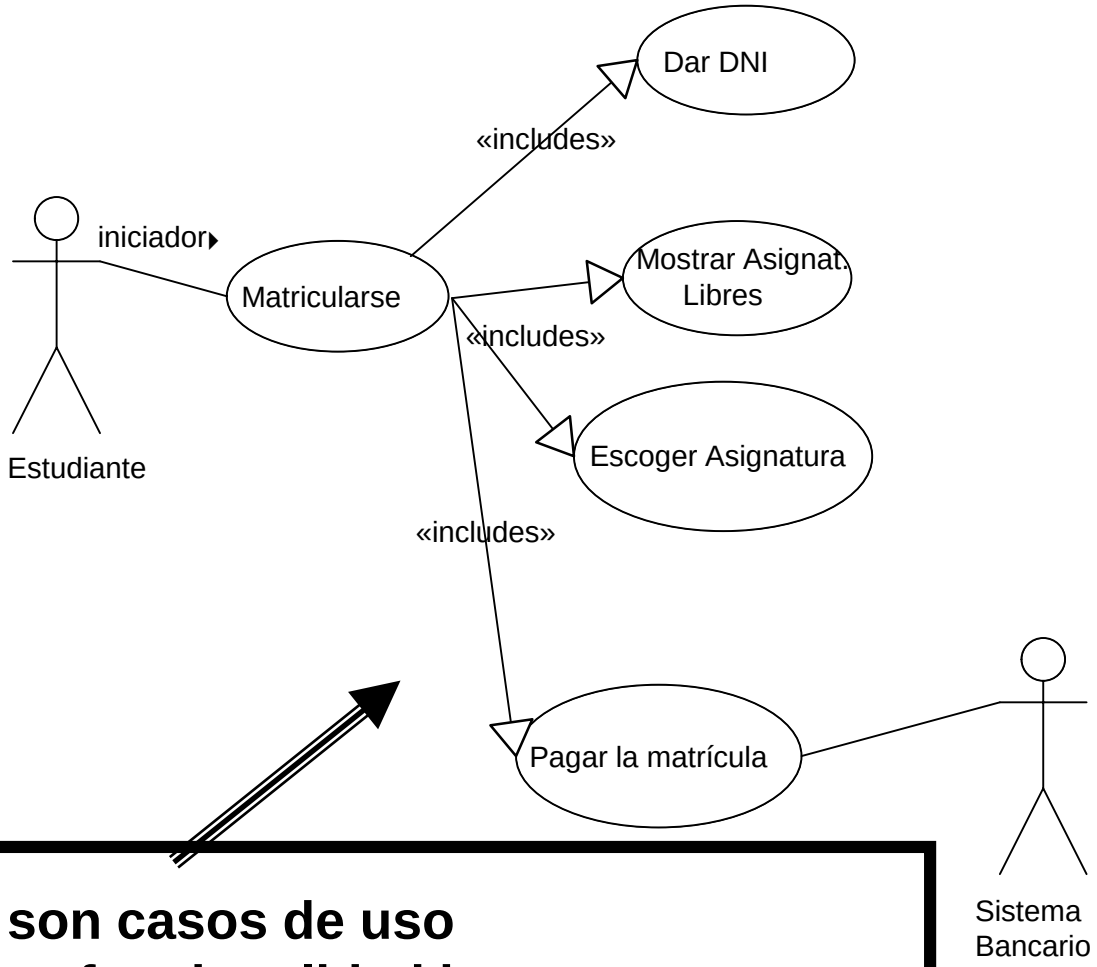

Asignar Pista
Asignar Tasifa


Mostrar Horas


Menu

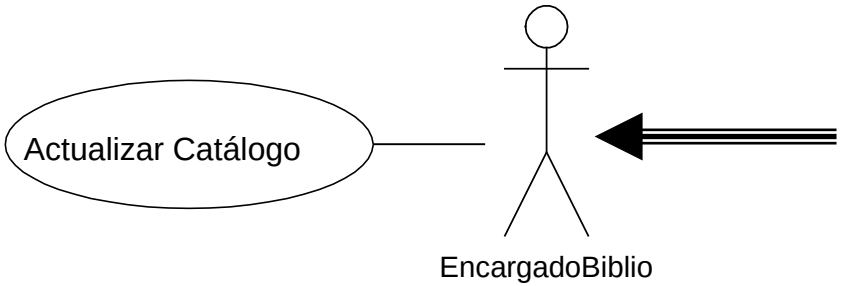
5) Error típico: realizar una descomposición funcional.

- Flujo de eventos
  - El estudiante proporciona su dni.
  - El sistema muestra las asignaturas en las que se puede matricular, si están libres todavía.
  - El estudiante escoge las asignaturas que desee.
  - El sistema calcula el precio de la matrícula y se realiza el pago en la cuenta del alumno por medio del sistema bancario.



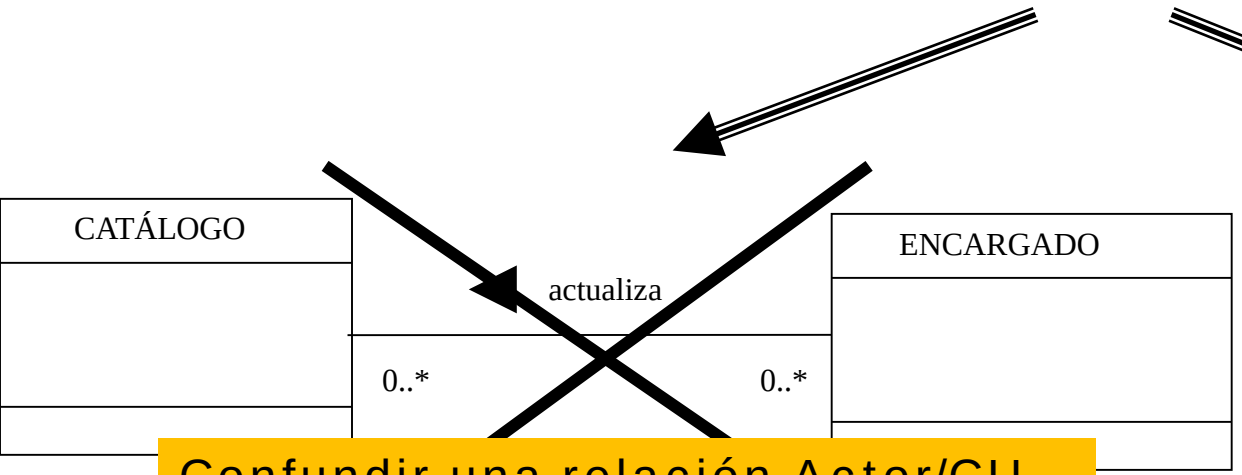
No son casos de uso
- sino funcionalidad interna
-que no puede ser ejecutada de manera independiente por ningún actor

6) Error típico: Confundir funcionalidad con datos



SIGNIFICA: El actor EncargadoBiblio puede ejecutar el caso de uso Actualizar Catálogo

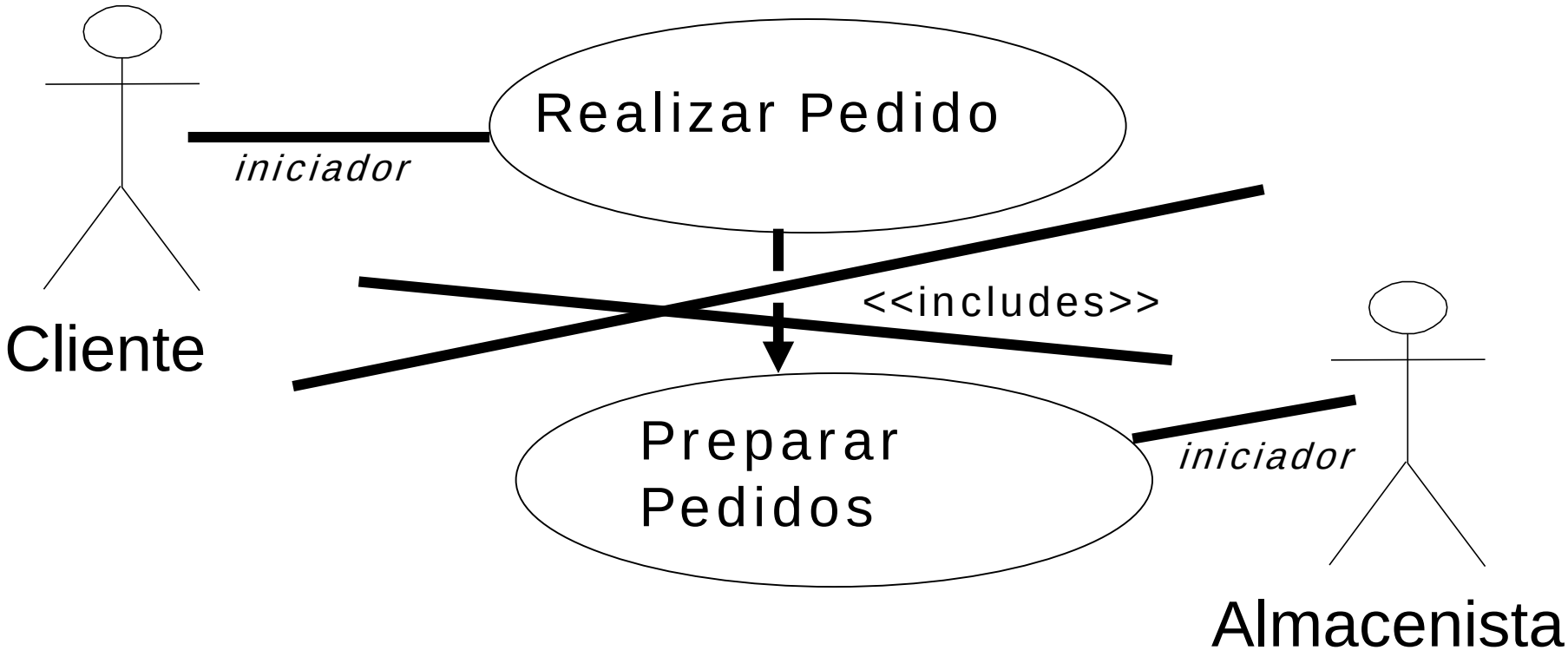
SIGNIFICA: SE DESEA ALMACENAR información sobre quién y/o que se ha actualizado el catálogo



Confundir una relación Actor/CU con una asociación entre clases

Confundir un CU con una clase

7) Errores típicos en CU: relación incorrecta entre casos de uso

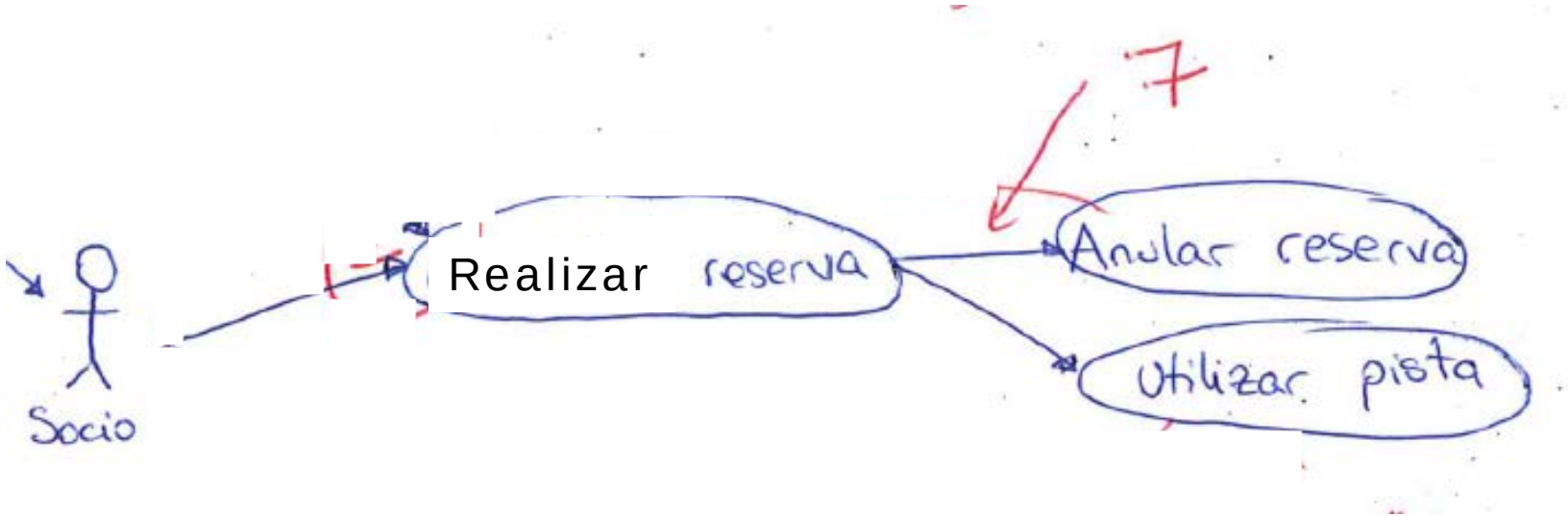


AMBOS son C.U. Después de realizar un pedido, el almacenista lo preparará para enviarlo al cliente, PERO NO SE HACE EN EL MOMENTO DE REALIZAR EL PEDIDO (el <<includes>> ESTÁ MAL).
El almacenista decidirá ejecutar el CU “Preparar Pedidos”
Flujo de Eventos de “Realizar Pedido”

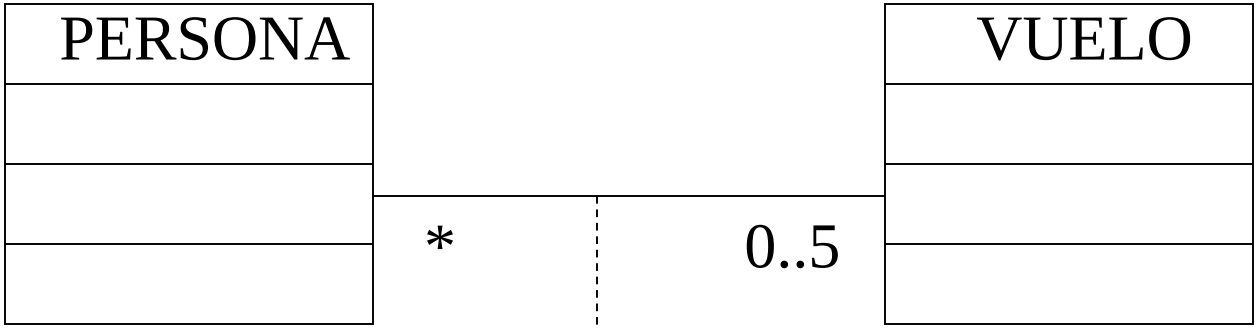
→ registrará el pedido

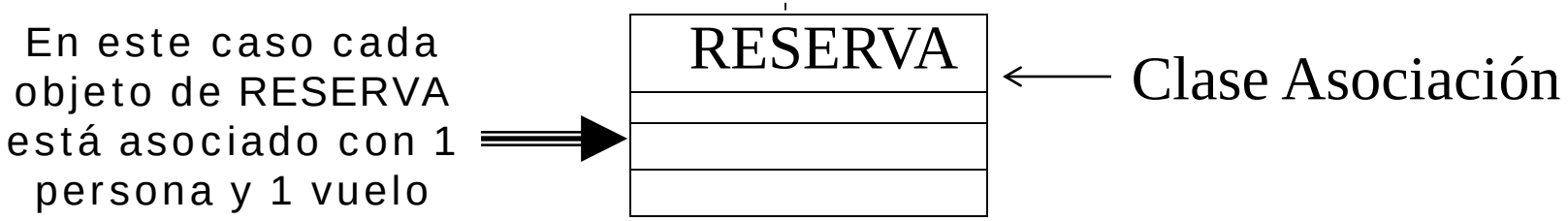
F.Eventos de “Preparar Pedidos”

→ obtendrá pedidos registrados y todavía no enviados

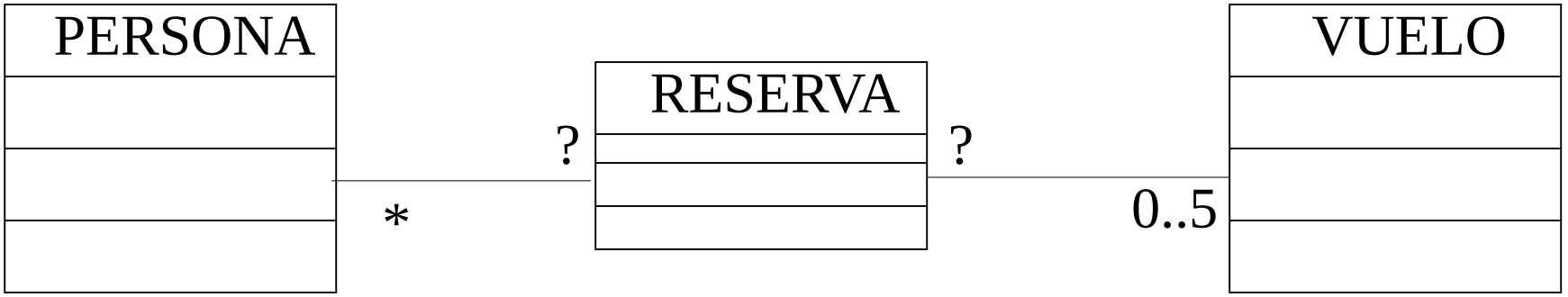

7
Realizar reserva
Anular reserva
Utilizar pista
Socio

8) Error típico: usar clase asociación en vez de clase con asociaciones

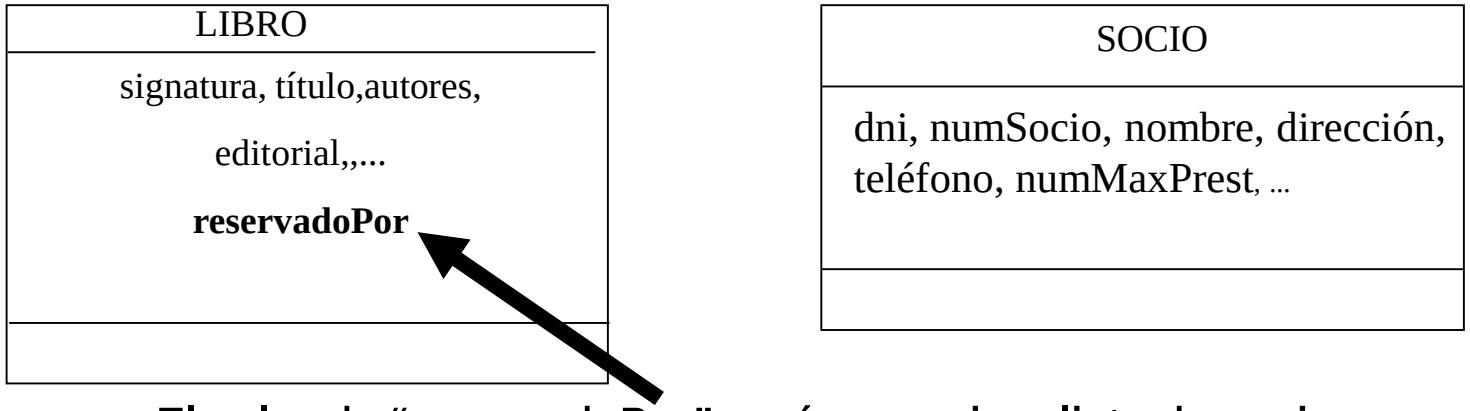




Si se quisiera reflejar que una reserva puede ser para VARIAS personas y que lo es de 0 a 5 vuelos habría que modelarlo así:

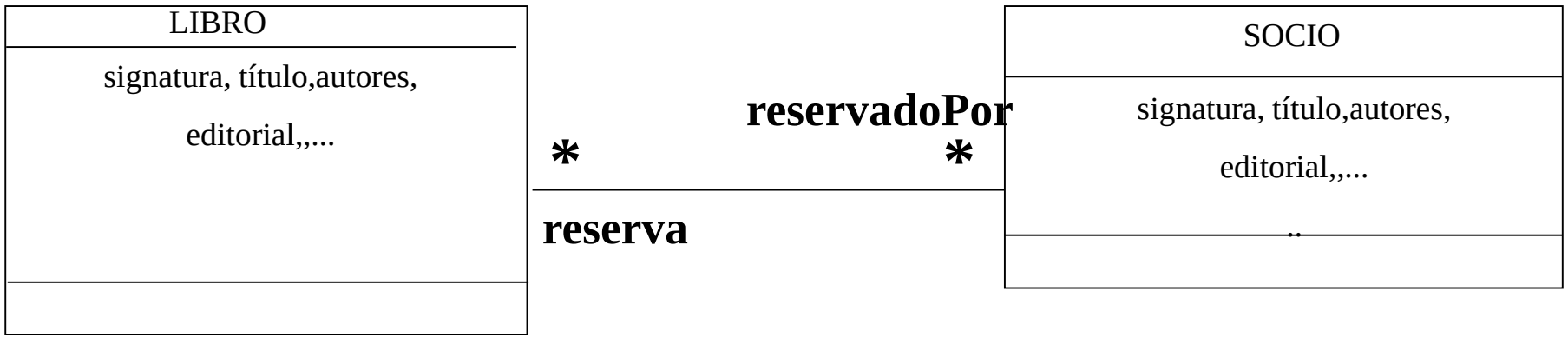


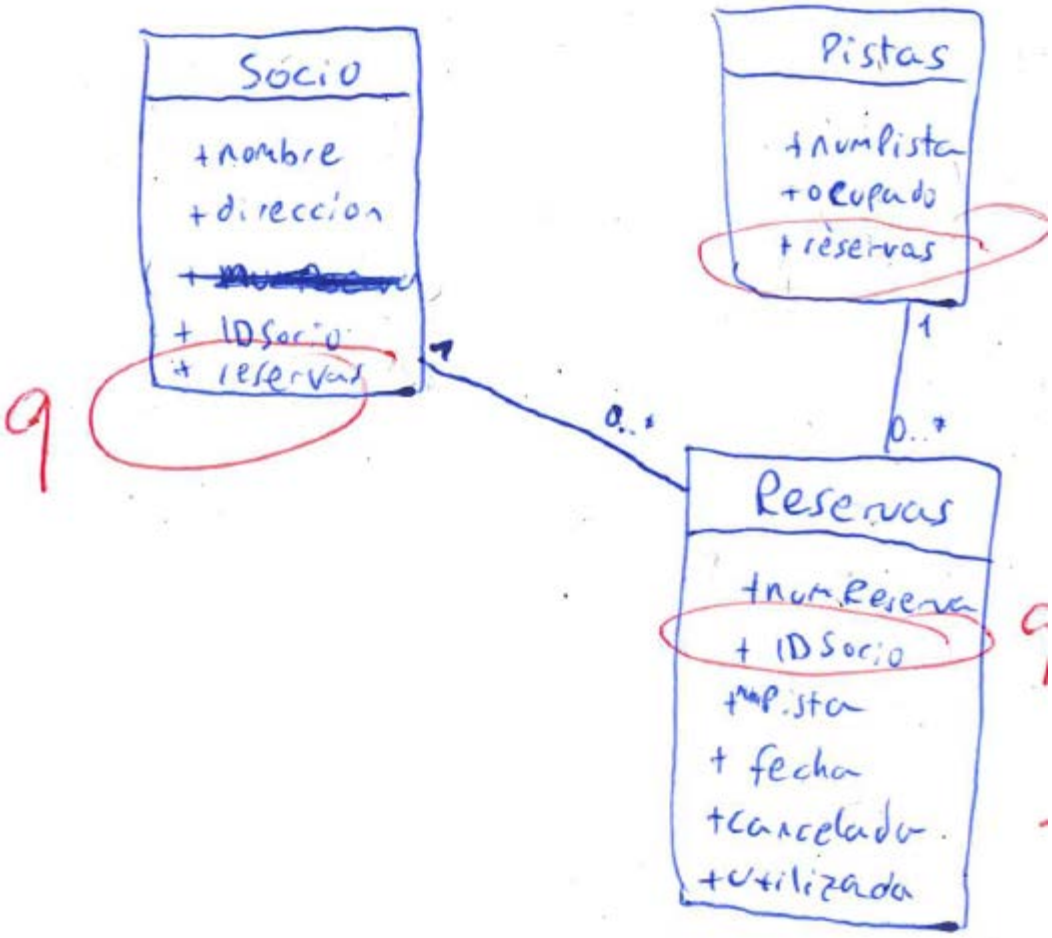
9) Error típico: usar atributos en vez de asociaciones



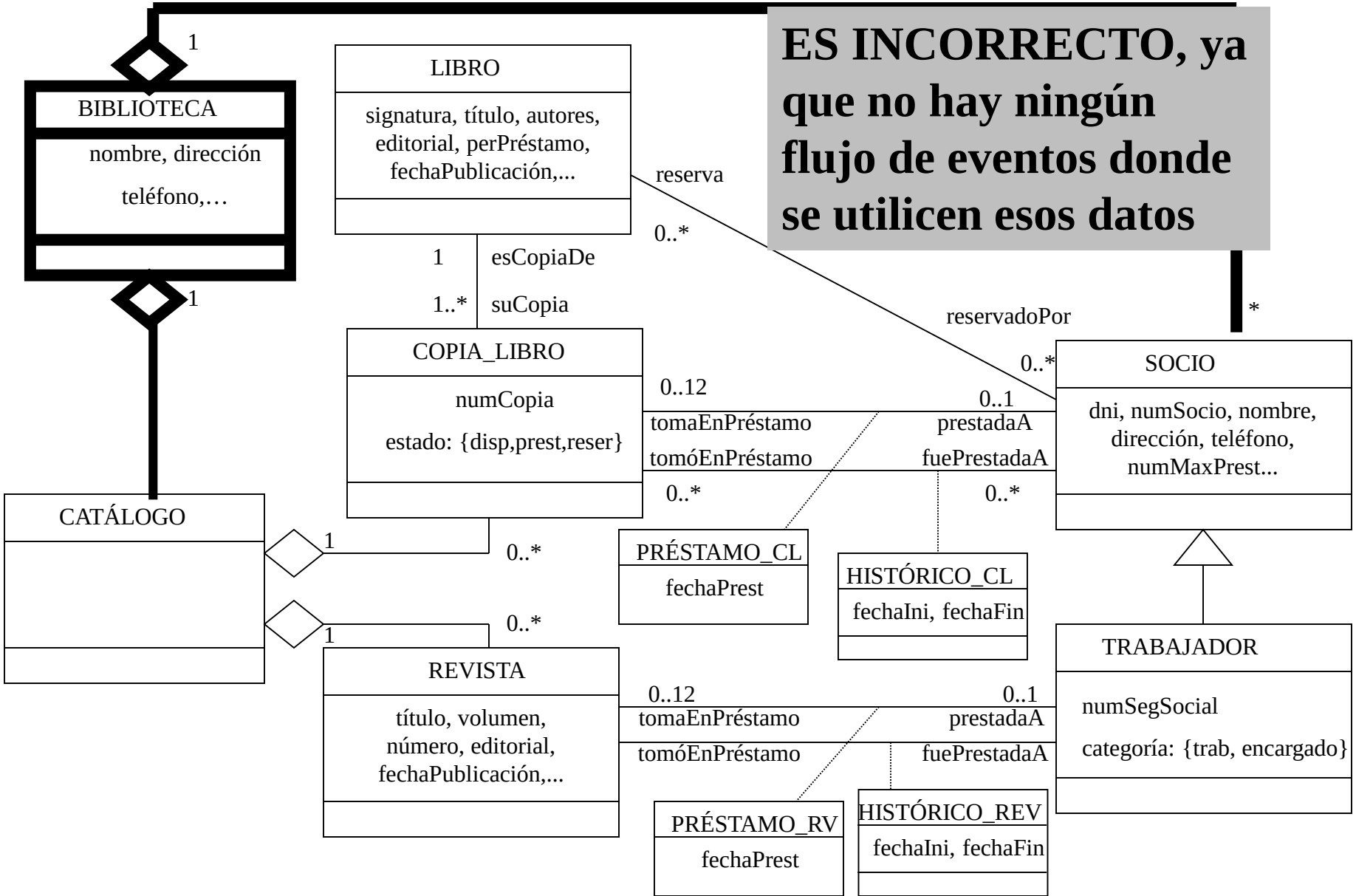
El valor de “reservadoPor” será un socio o lista de socios…
PERO, ya tenemos la clase SOCIO en el MODELO DEL DOMINIO

HAY QUE HACERLO USANDO ASOCIACIÓN




Socio
+nombre
+direccion
+ IDSocio
+ reservas
Pistas
+numPista
+ocupado
+reservas
1
0..*
0..*
Reservas
+numReserva
+ IDSocio
+numPista
+ fecha
+cancelada
+utilizada

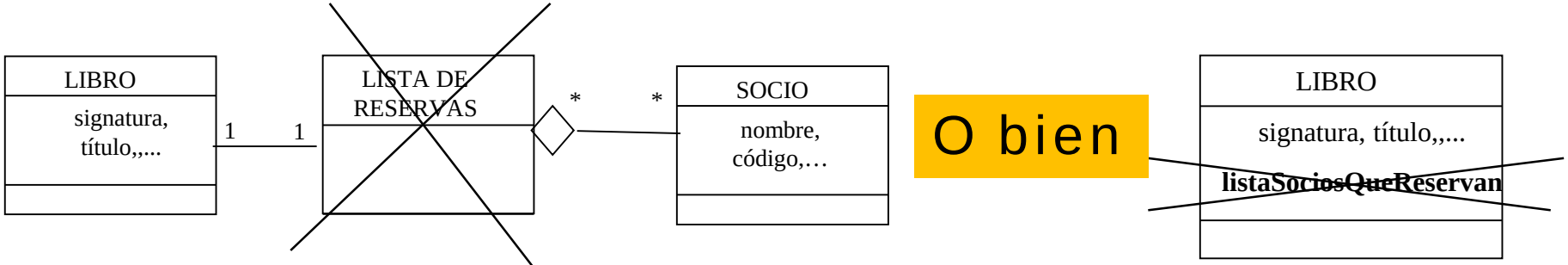
10) Error típico: añadir elementos no necesarios según Flujo de Eventos



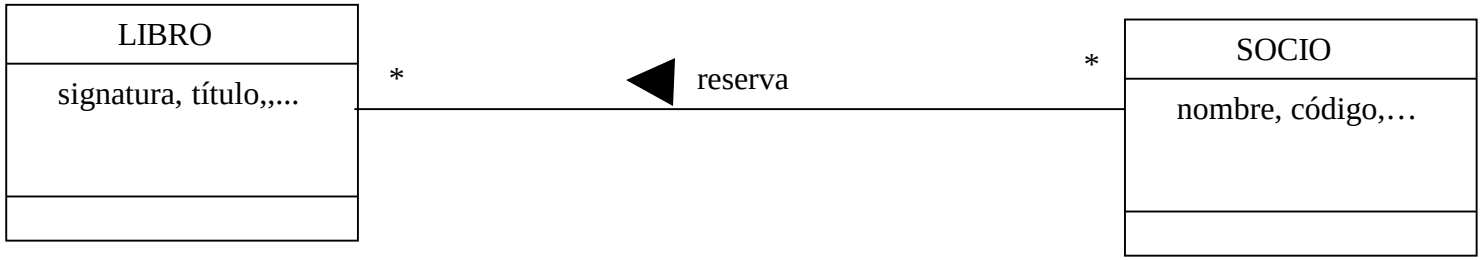
ES INCORRECTO, ya que no hay ningún flujo de eventos donde se utilicen esos datos

11) Error típico: añadir clases que no son conceptuales (o que son más de diseño/implementación)

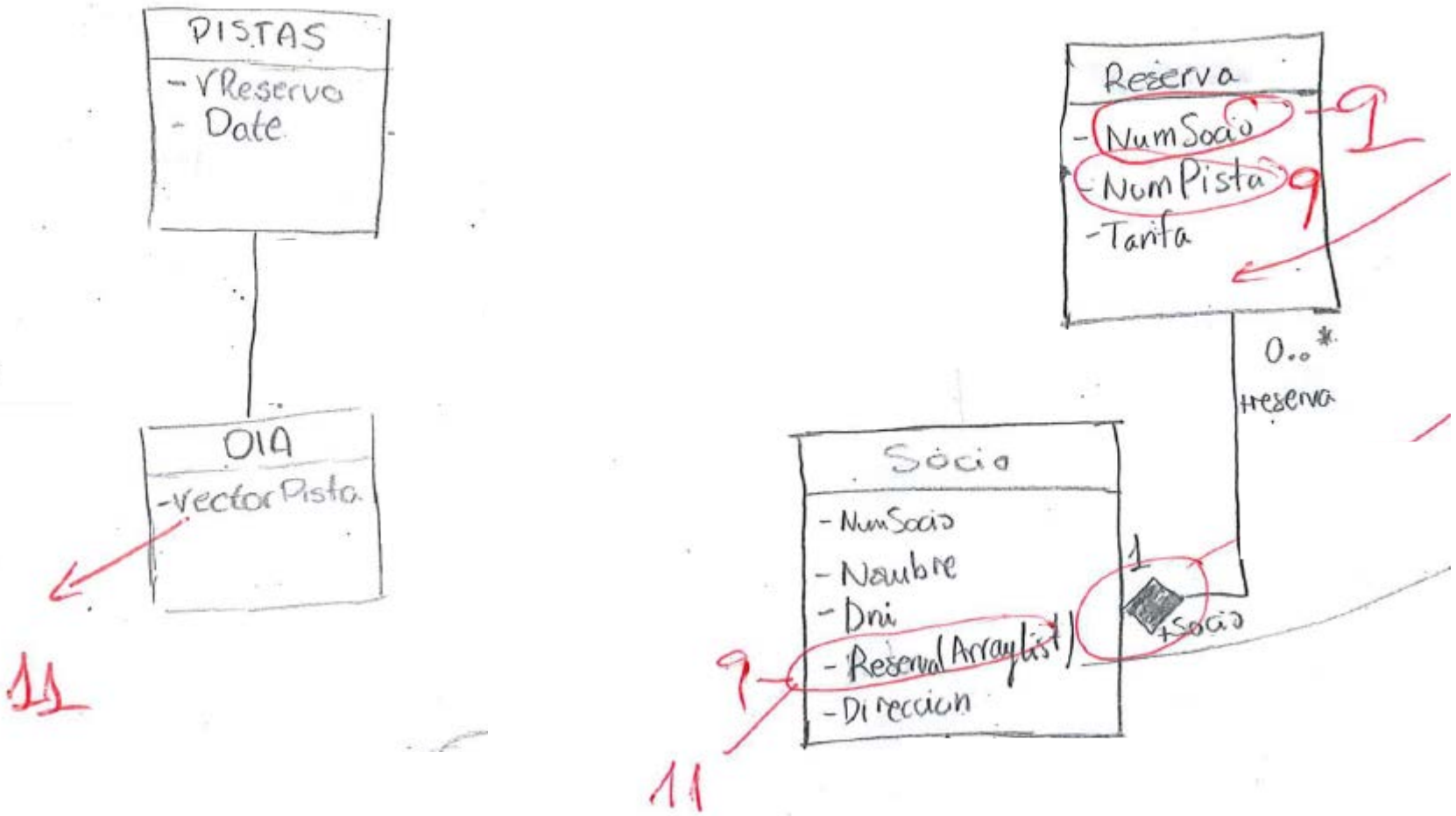
Requisitos: Los socios podrán reservar libros. Por cada libro, habrá una lista de reservas con los socios interesados.



Para modelar el requisito basta una asociación:



En el modelo del dominio no hay por qué decidir si usaremos una LISTA, o una TABLA HASH, o lo que sea.
Ya se decidirá durante el DISEÑO y/o IMPLEMENTACIÓN

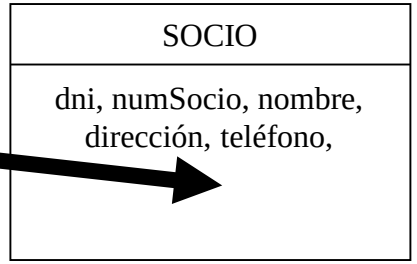

PISTAS
- VReserva
- Date
DIA
-vectorPista
11
Reserva
-NumSocio
-NumPista
-Tarifa
1
9
9
0..*
+reserva
Socio
-NumSocio
-Nombre
-Dni
-Reserva(ArrayList)
-Direccion
1
+Socio
9
11

12) Error típico: falta de coherencia entre MCU y MD

Hay datos en los FLUJOS DE EVENTOS que no existen en el Modelo del Dominio.

F.Eventos de “Tomar Préstamo Copia Libro”
-…
- se comprueba que el socio no se pasa del <u>núm. máximo de préstamos</u>

NO EXISTE:
numMaxPrest...



Hay clases, atributos o asociaciones que no se usan en los FLUJOS de EVENTOS.

F.Eventos 1 …
…

F.Eventos 2 …
…

En ningún sitio de los
Flujos de Eventos
SE UTILIZA EL EMAIL

12 → Sin flujos de eventos, seguro.

FLUJO DE EVENTOS

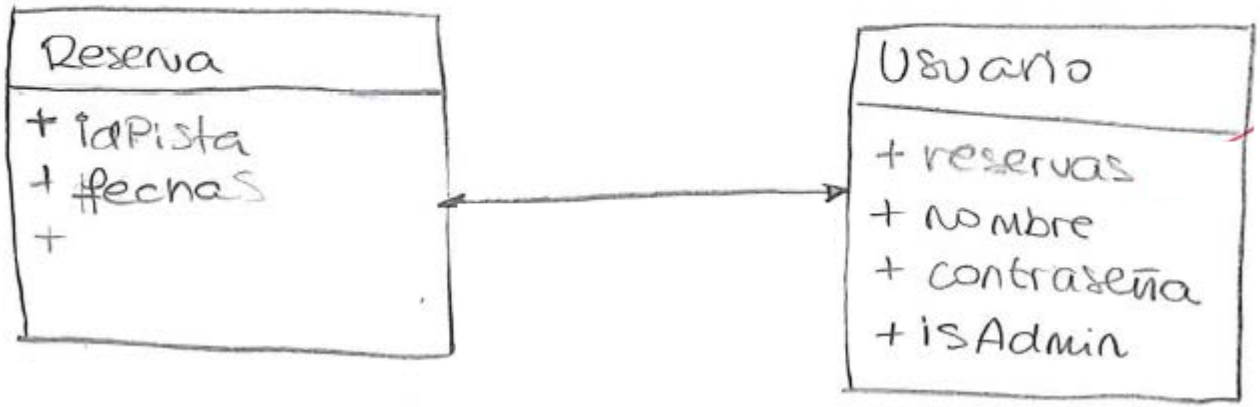
RESERVAR

Flujo Basico

12

1. *System* muestra las pistas disponibles con fecha y hora.

UML



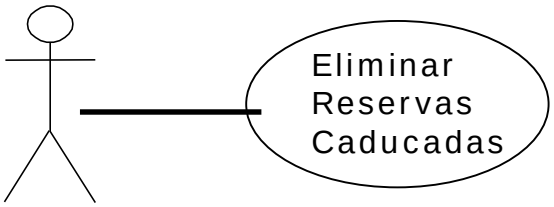
13) Error típico: falta funcionalidad requerida

Las reservas se guardarán como máximo durante 5 días

F.Eventos de “Reservar”
-... “hacer lo necesario para reservar”
- Si han pasado 5 días sin que se compre lo reservado, se eliminará la reserva

El problema es que en el mismo CU no se puede hacer la reserva y esperar 5 días a ver si compra lo reservado. El CU debe terminar rápidamente para comunicar al usuario si tiene la reserva o no.

Nuevo CU:



F.Eventos: “Eliminar Reservas Caducadas”
-El sistema mira todas las reservas activas y si han pasado 5 días desde que se hizo la reserva, se eliminan
Req. No Funcional:
-Se ejecuta diariamente, según haya sido configurado por el administrador

- Imprimir facturas
- Anotar NO utilización de pista (para tarifa T3)
  - O bien “Anotar utilización de pista” y controlar después, en otro caso de uso cuáles no se han utilizado para ponerle tarifa T3 (por ejemplo en “imprimir facturas”)

14) Error típico: faltan clases/asociaciones/atributos requeridos

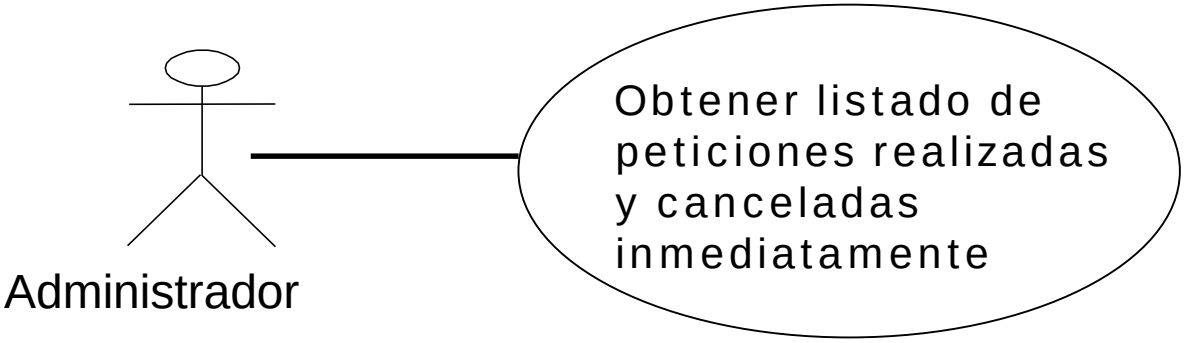
Cada socio tiene un número máximo de reservas de libros que puede hacer (depende de cada socio) y se debe guardar la fecha en la que pide reservar cada libro

Falta atributo “numMaximoReservas” en la clase SOCIO

Falta clase asociación con atributo “fechaReserva” entre las clases LIBRO y SOCIO (o bien una clase con dos asociaciones)

- Facturas
- Reservas

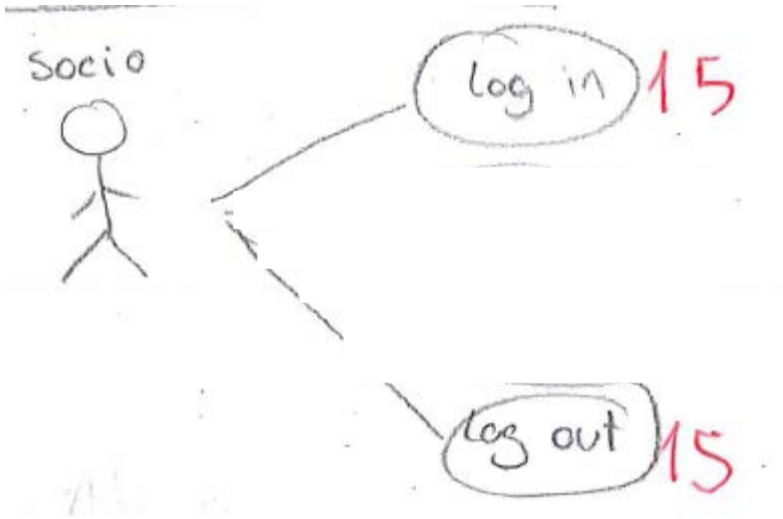
15) Error típico: añadir funcionalidad no solicitada



No se ha solicitado expresamente esa funcionalidad, por lo que no puede haber un actor interesado en la misma.

Los desarrolladores de software, por supuesto, pueden pensar en funcionalidades que consideren de interés que los clientes pueden no haberse planteado (por desconocimiento de la tecnología, porque no se les había ocurrido,...).

Pero finalmente, la decisión de que esa posible funcionalidad sea o no un caso de uso será de los clientes; deberá haber algún actor interesado en la misma.



Seguro que serán necesarios, pero para nuestro “filtro” mejor no perder el tiempo

16) Error típico: no especificar cuándo se ejecuta la funcionalidad

No se ve claramente en qué momento se puede ejecutar el caso de uso, tiene una ejecución retardada (cuando ocurra algo). Los casos de uso deben terminar en un tiempo razonable, para informar al actor del resultado.

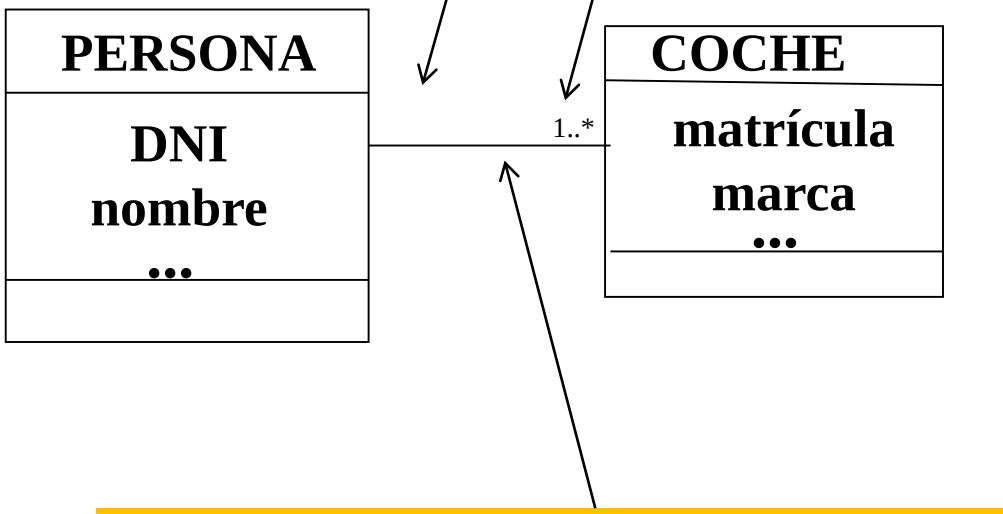
El caso “Reservar” anterior (error 13) tenía ese problema:

F.Eventos de “Reservar”
-... “hacer lo necesario para reservar”

- Si han pasado 5 días sin que se compre lo reservado, se eliminará la reserva

17) Error típico: problemas de cardinalidades y nombres apropiados en el modelo del dominio

FALTA CARDINALIDAD: ¿1?

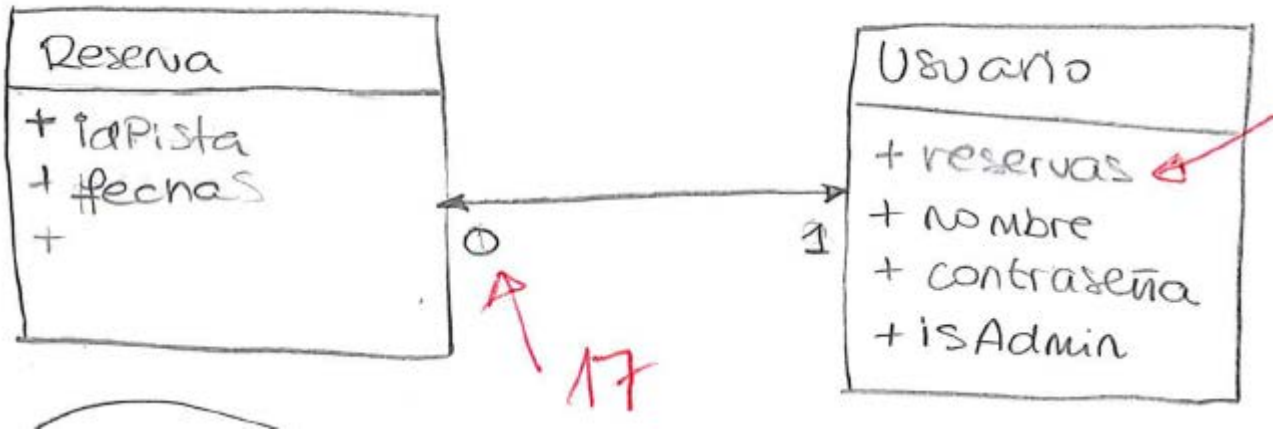


CARDINALIDAD INCORRECTA:

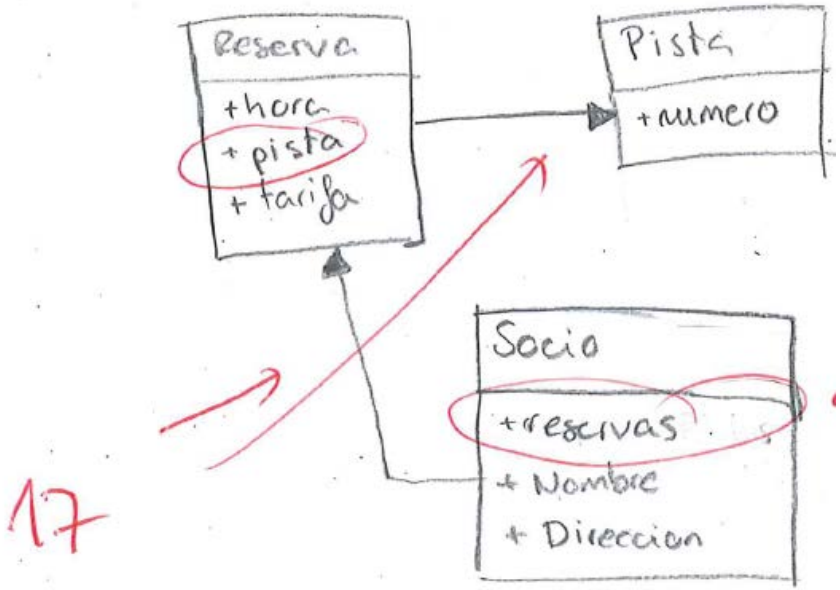
Si no queremos obligar a que en el momento de insertar una nueva persona haya que meter necesariamente uno de sus coches, sino que se pueda hacer más tarde entonces la cardinalidad sería:

* = 0..*

FALTA NOMBRE DE LA ASOCIACIÓN: ¿alquila? ¿posee? ¿haTenidoAccidenteCon?



MODELO DE DOMINIO



18) Error típico: detalles innecesarios de la interfaz gráfica de usuario en el FLUJO de EVENTOS

F.Eventos de “Matricularse”
-El sistema muestra en un JComboBox los códigos de todas las asignaturas
- El estudiante selecciona una asignatura y pulsa el botón para formalizar la matrícula

En el flujo de eventos no habría que decidir todavía si se usa un JComboBox y botones (confuso para cliente)

Mejor así:

F.Eventos de “Matricularse”
-El sistema muestra los códigos de todas las asignaturas
- El estudiante selecciona una asignatura y solicita la matrícula

Lo que SÍ es RECOMENDABLE proporcionar aparte el prototipo de interfaz donde pueden aparecer dibujados JComboBox y botones (podría ayudar al cliente)

Hacer Reserva

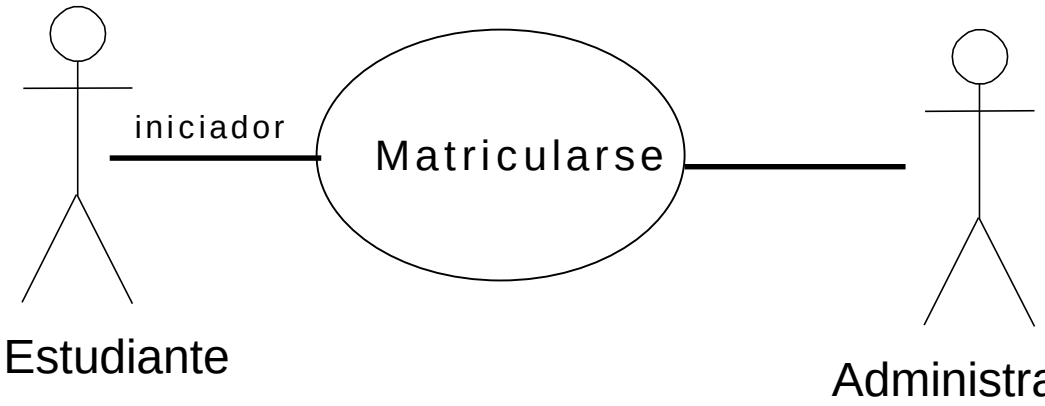
Flujo Básico:

- El sistema muestra la ventana principal.
- El usuario selecciona la opción "hacer reserva".
- El sistema muestra la ventana de reservas.

18

19) Error típico: problemas con actores que proporcionan/reciben datos (o no) en los FE

F.Eventos de “Matricularse”
-El sistema muestra los códigos de todas las asignaturas
- El estudiante selecciona una asignatura y pulsa el botón para formalizar la matrícula

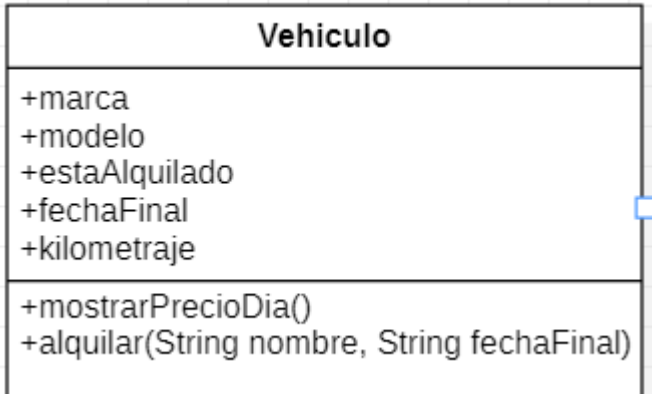


El administrativo NO APARECE en el FE del CU, por lo que NO ES NECESARIO (aunque sea el que introduce los datos)

El actor Administrativo sería necesario si apareciera en el FE: “El administrativo valida la matrícula”

20) Error típico: poner métodos en las clases del dominio

Durante la captura de REQUISITOS se va definiendo el modelo del dominio. En el mismo SÓLO deben aparecer CLASES con sus ATRIBUTOS, asociaciones y clases asociación

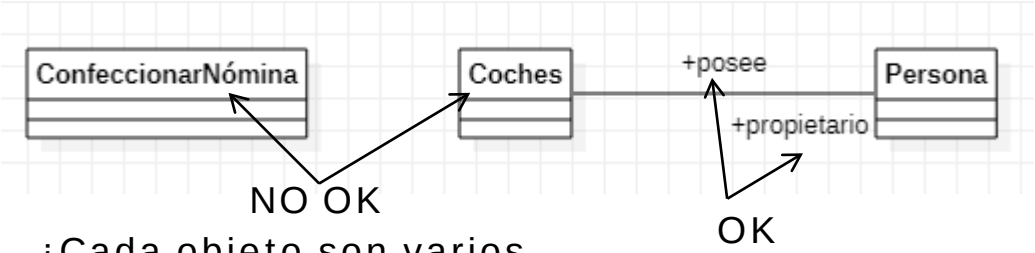


Durante el DISEÑO se irán identificando nuevas CLASES junto con sus métodos. Así se definirá el DIAGRAMA DE CLASES del DISEÑO.

21) Error típico: nombres incorrectos para CU, clases, asociaciones,...

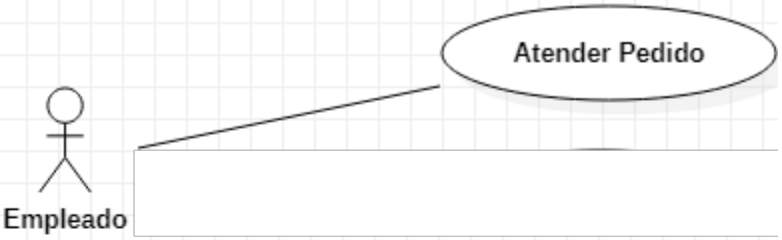
CU: VERBO + COMPLEMENTO DIRECTO
Relacionado con el DOMINIO/CONTEXTO
Excepciones: si son muy conocidos y genéricos Login, Registro...

CLASES: sustantivos/nombres en singular
ASOCIACIONES: verbos, pero podrían ser sustantivos en los extremos



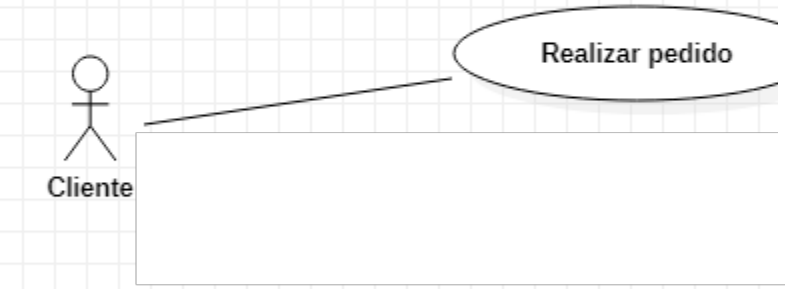
¿Cada objeto son varios Coches, o es solo uno?

22) Error: definir casos de uso diferentes cuando se trata del mismo caso de uso



Flujo de eventos: Atender pedido

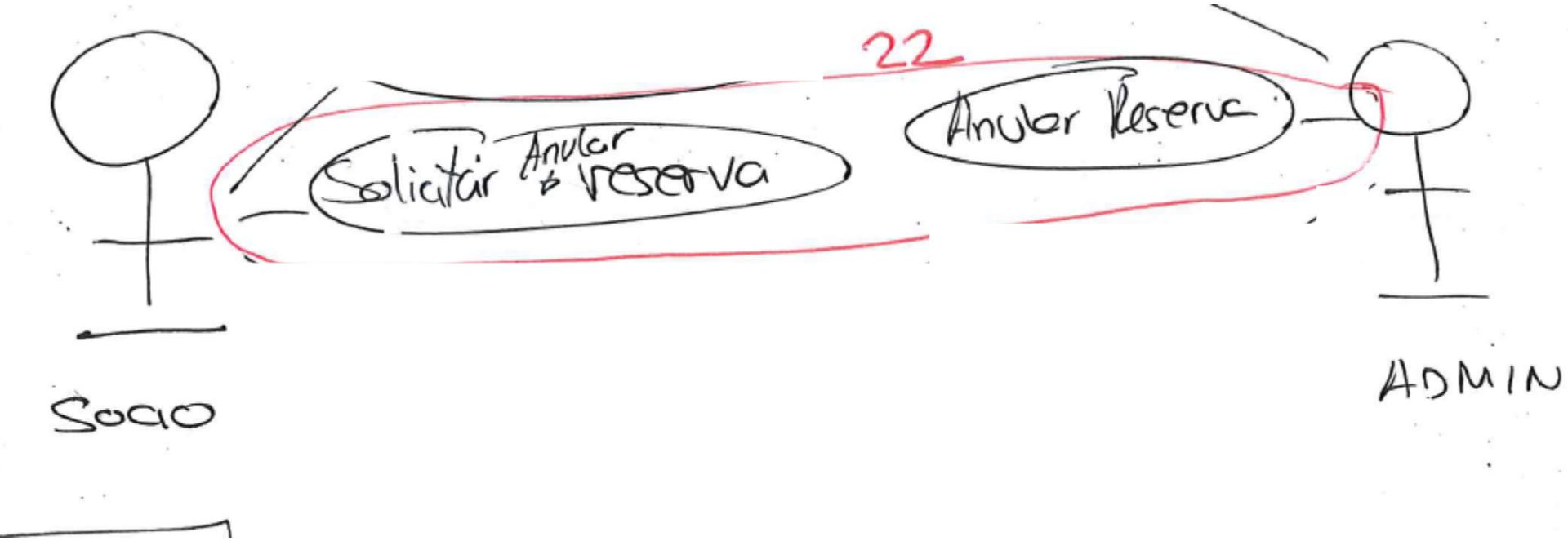
1. El empleado atiende una llamada de telefono para atender un pedidio de un cleinte.
2. El empleado decide preguntarse por sus datos personales del registro, para comprobar que e cliente está registrado.
3. El empleado le pregunta al cliente de que tamaño quiere la pizza.
4. El empleado le pregunta al cleinte que ingredientes quiere para la pizza.
5. El empleado comprueba la disponibilidad de dichos ingredientes.



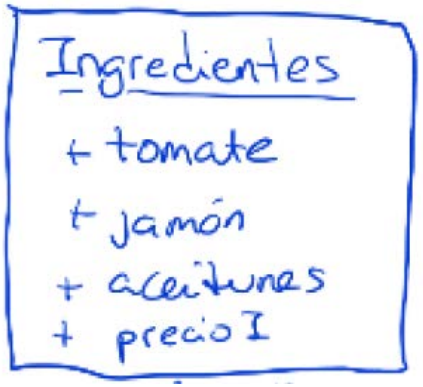
Flujo de eventos: Realizar pedido

1. El cliente llama por teléfono a PIZZASTEL donde un empleado les tomará nota.
2. El cliente facilitará al empleado sus datos personales del registro para comprobar que es un cliente registrado.
3. El cliente le dice al empleado la pizza que quiere con sus correspondientes ingredientes, y el número de latas en caso de que desee pedir alguna.

Es el mismo CU “Realizar Pedido” asociado con los 2 actores


22
Solicitar Anular reserva
Anular Reserva
SOCIO
ADMIN

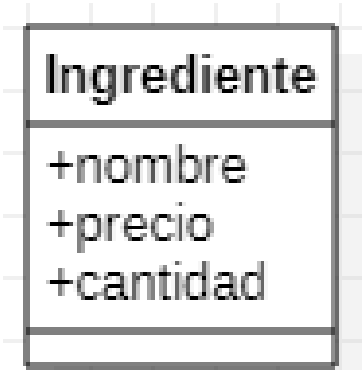
23) Error: definir atributos de clases en vez de objetos



Posibles objetos de esta clase:
@Ing1 <2,3,1,8.5€>
@Ing2 <1,2,1,6€>

Con precios tomate=0.5, jamón=2, aceitunas=1.5

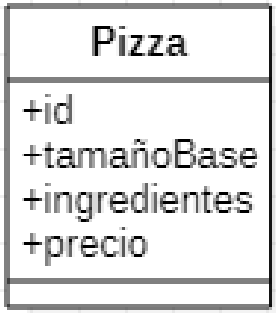
Si se quisiera añadir un nuevo ingrediente (curry, con precio=0.75), habría que modificar el esquema de clases, lo cual haría muy poco extensible la aplicación actual



Tendríamos objetos
@I1 <“tomate”,0.5,...>,...

Trabajar con el ingrediente curry, implicaría tener un nuevo objeto en la BD: @In <“curry”m0.75,...> y no modificar las clases:

24) Error: definir atributos multivaluados en vez de clases con asociaciones



Posibles objetos de esta clase:
@Pizza1 <1,“grande”, “queso,tomate”,8.5€>
@Pizza2 <2,“mediana”, “tomate,aceitunas”,7€>

Es correcto sólo si “ingredientes” se quiere tratar como un String.

Pero si se quiere realizar un procesamiento sobre los ingredientes como: “conocer el precio de los ingredientes”, “conocer el stock de un ingrediente”, “cuántas pizzas tienen el ingrediente queso”,... entonces en vez de modelar “ingredientes” COMO UN SIMPLE ATRIBUTO hay que definir una CLASE con su correspodiente ASOCIACIÓN