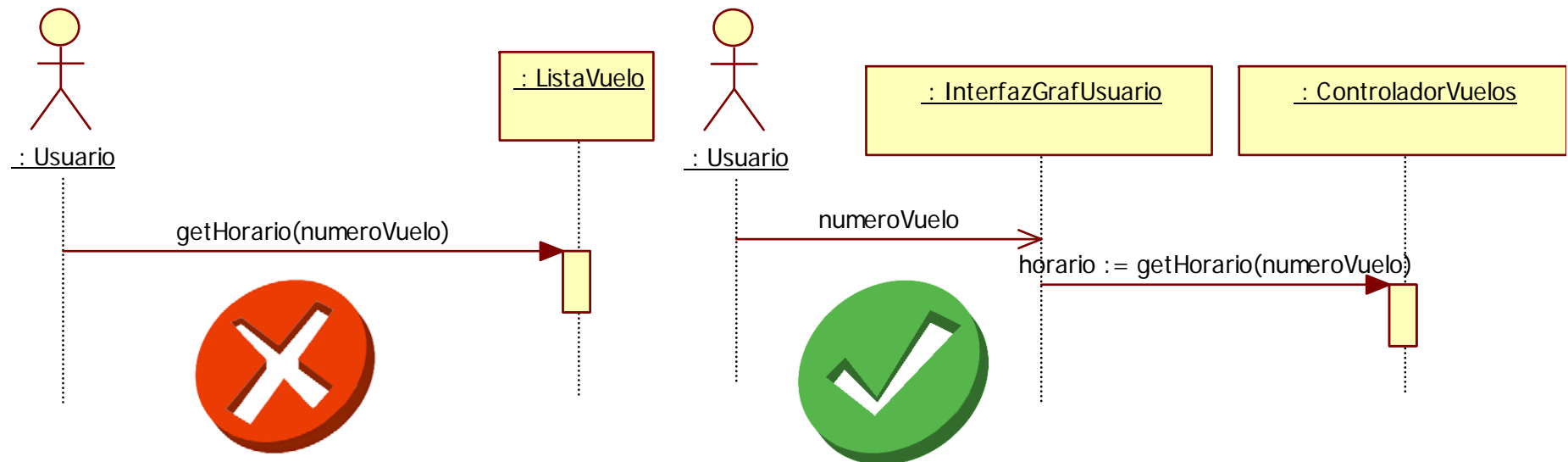


Errores típicos en diseño (y algunas cosas posibles)

Error típico: no utilizar una clase interfaz/frontera para interactuar con el actor

Flujo de Eventos:

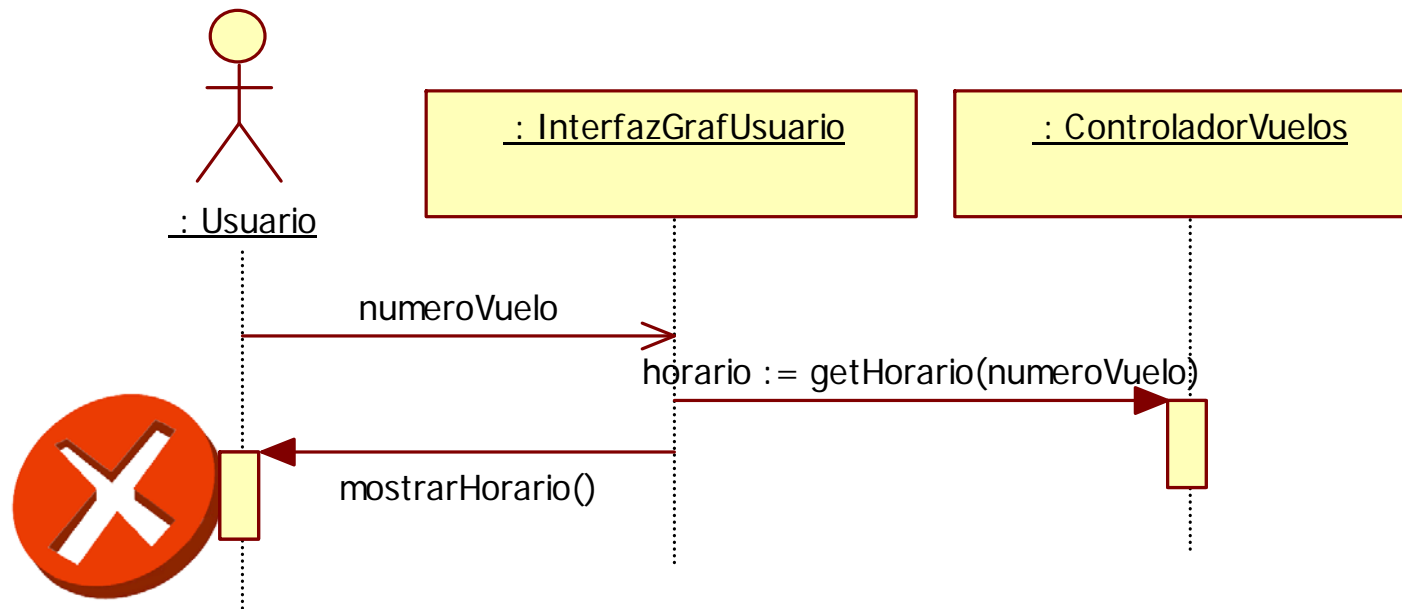
- 1.- El usuario proporciona un número de vuelo
- 2.- El sistema obtiene el horario de dicho vuelo y se le muestra al usuario



Error típico: solicitar que un actor ejecute un método

Flujo de Eventos:

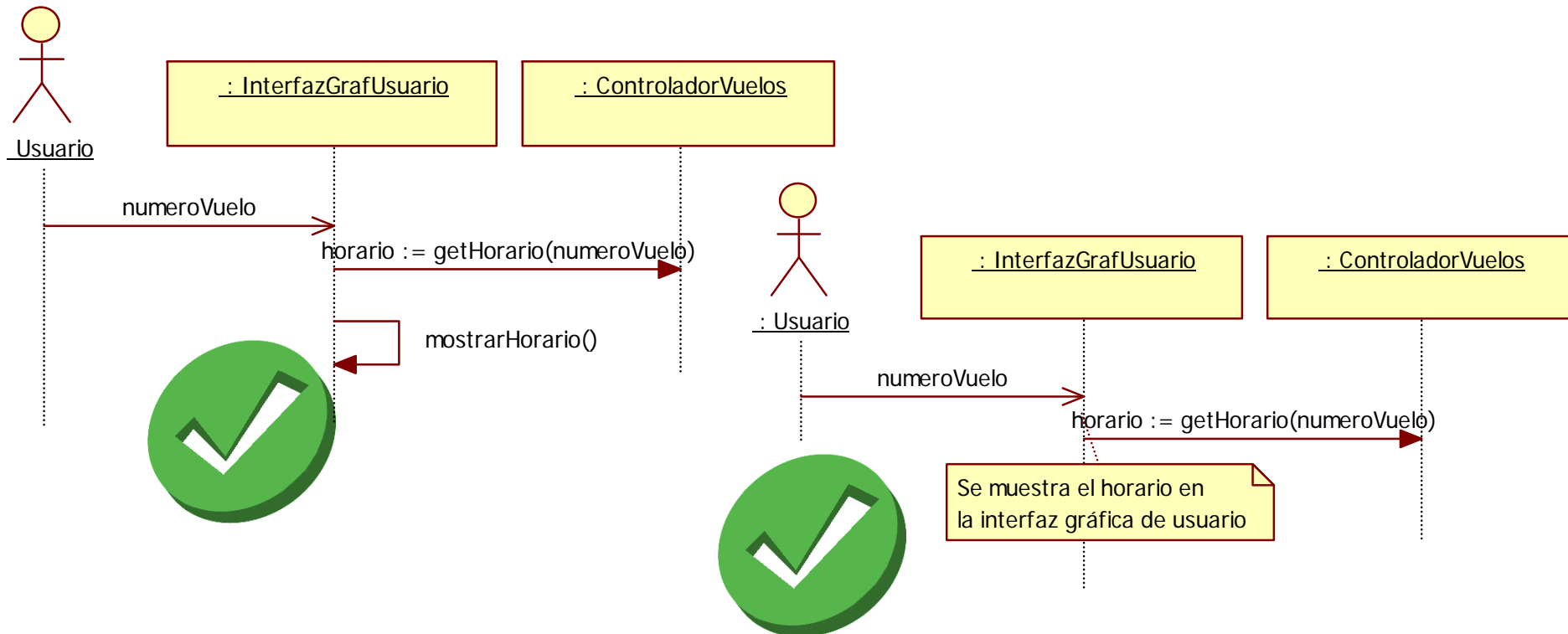
- 1.- El usuario proporciona un número de vuelo
- 2.- El sistema obtiene el horario de dicho vuelo y se le muestra al usuario



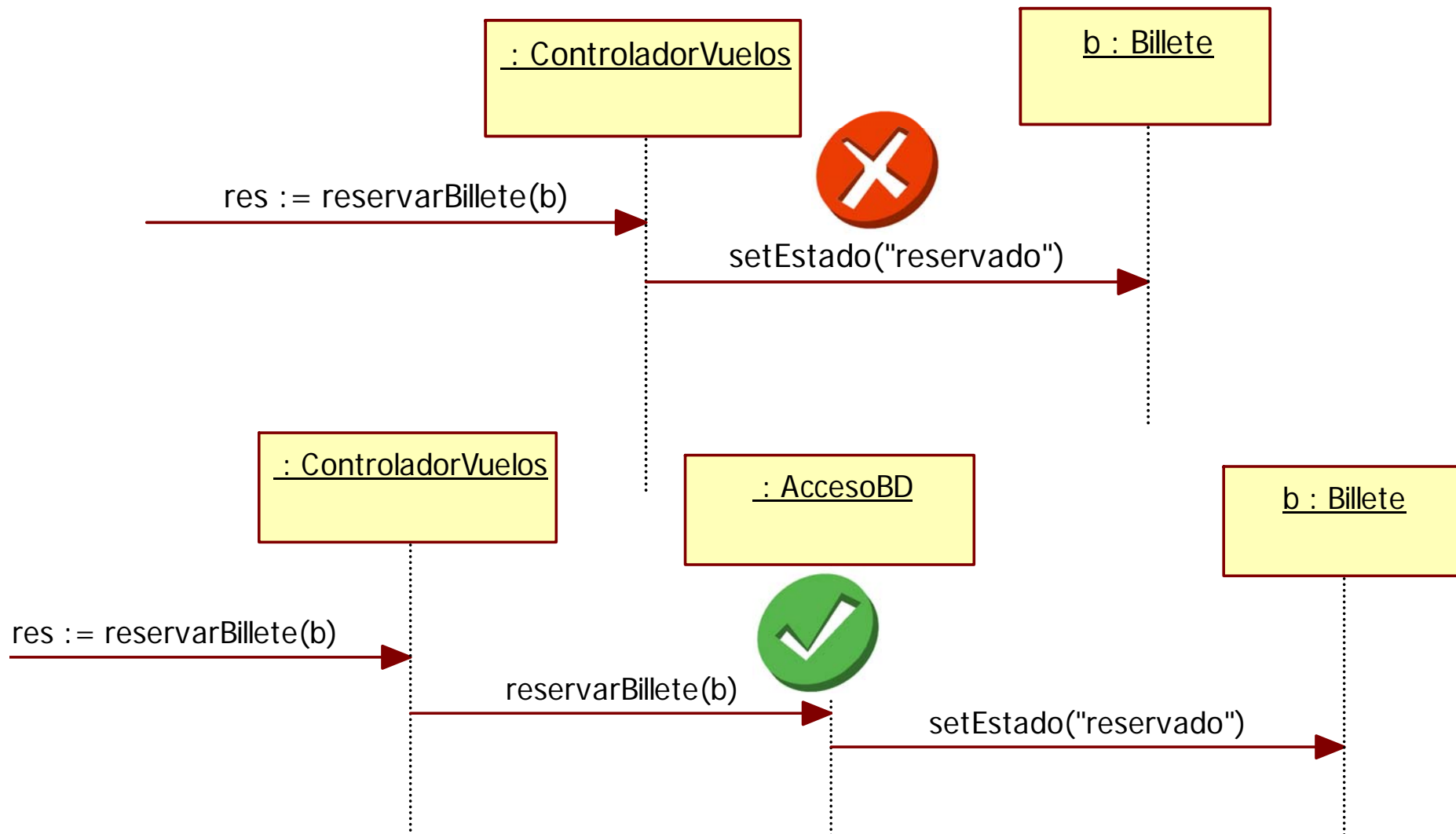
Error típico: solicitar que un actor ejecute un método

Flujo de Eventos:

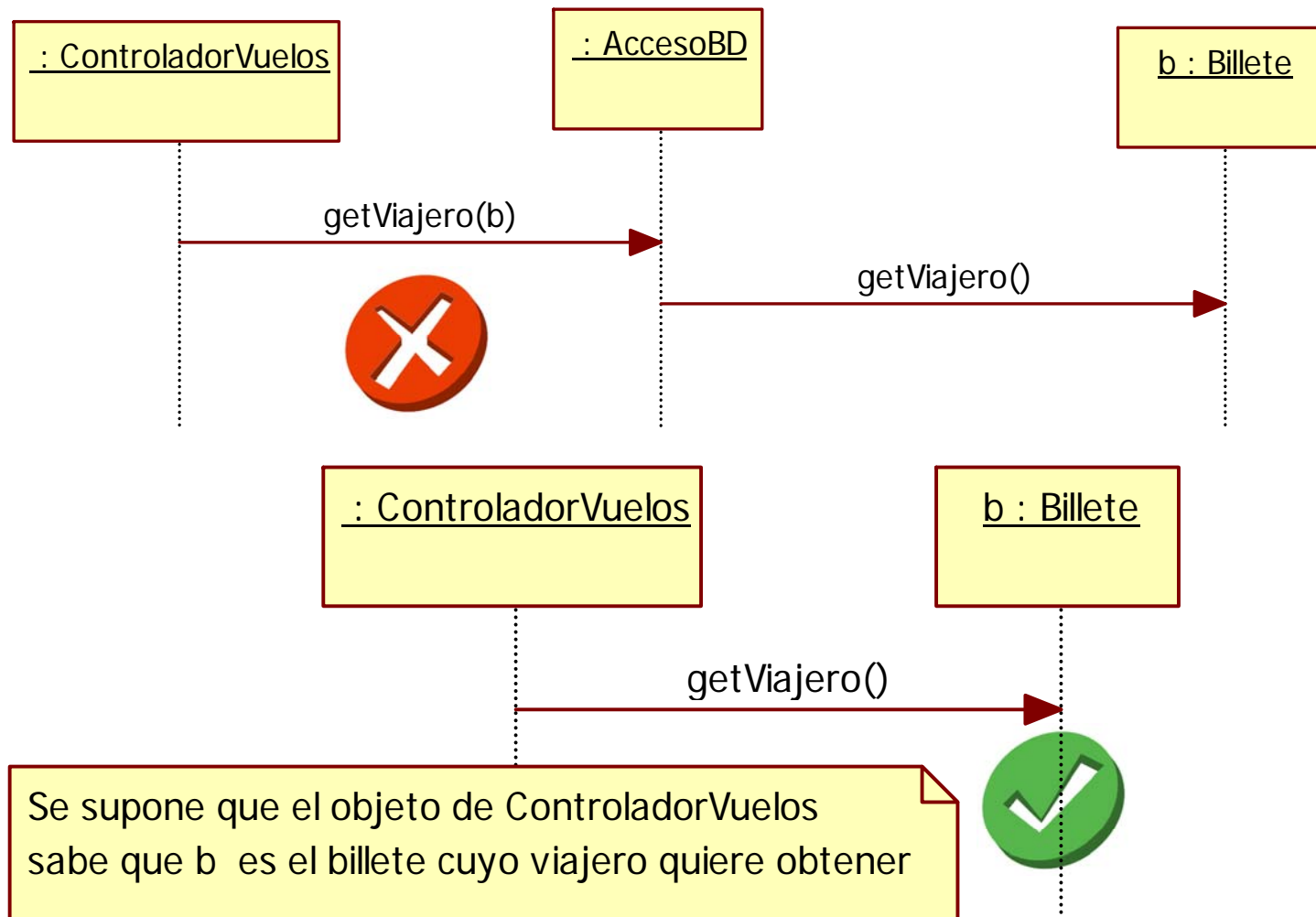
- 1.- El usuario proporciona un número de vuelo
- 2.- El sistema obtiene el horario de dicho vuelo y se le muestra al usuario



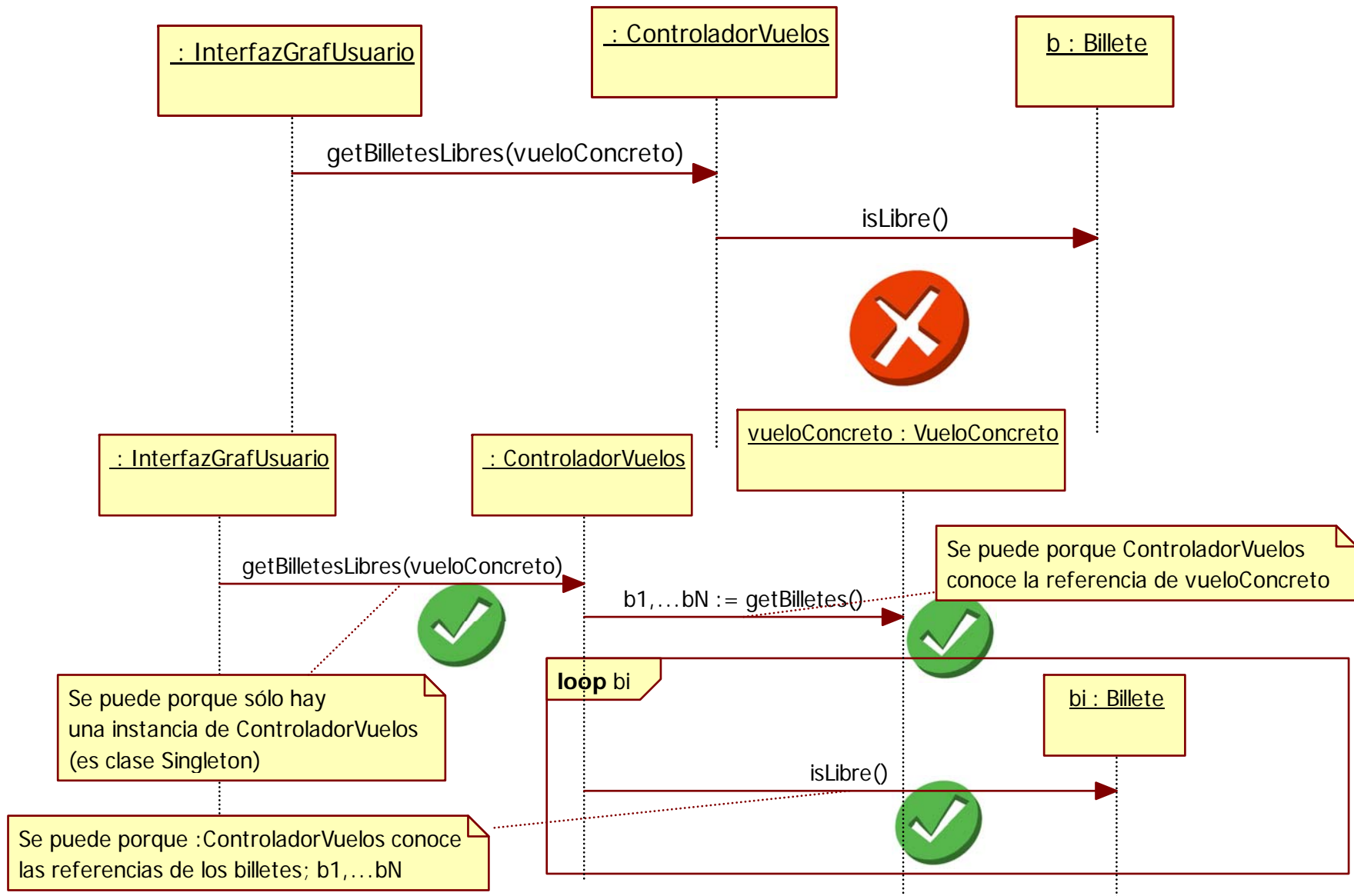
Error típico: realizar operaciones de actualización/borrado/inserción directamente en los objetos del dominio



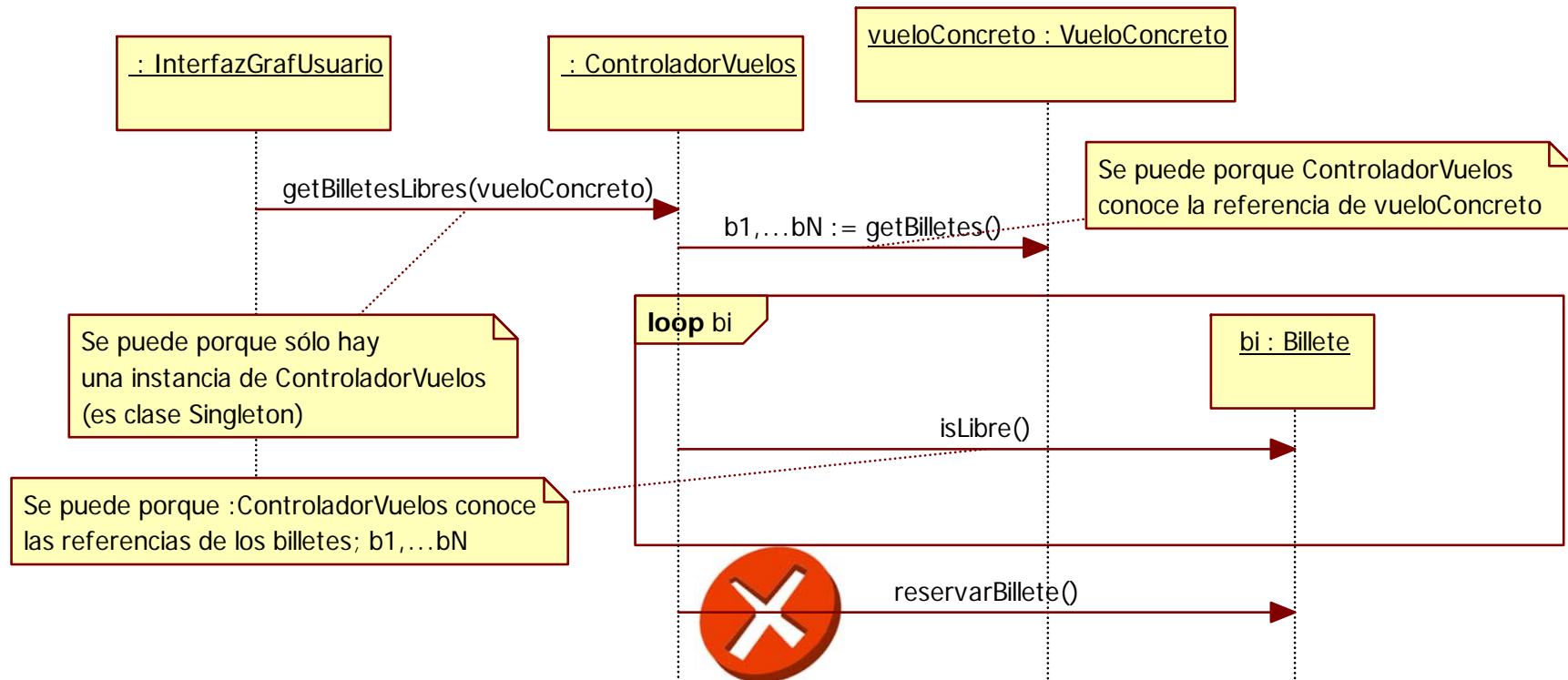
Error típico: consultar datos a través de la BD, conocido el objeto del dominio



Error típico: enviar un método a un objeto cuya referencia no es conocida

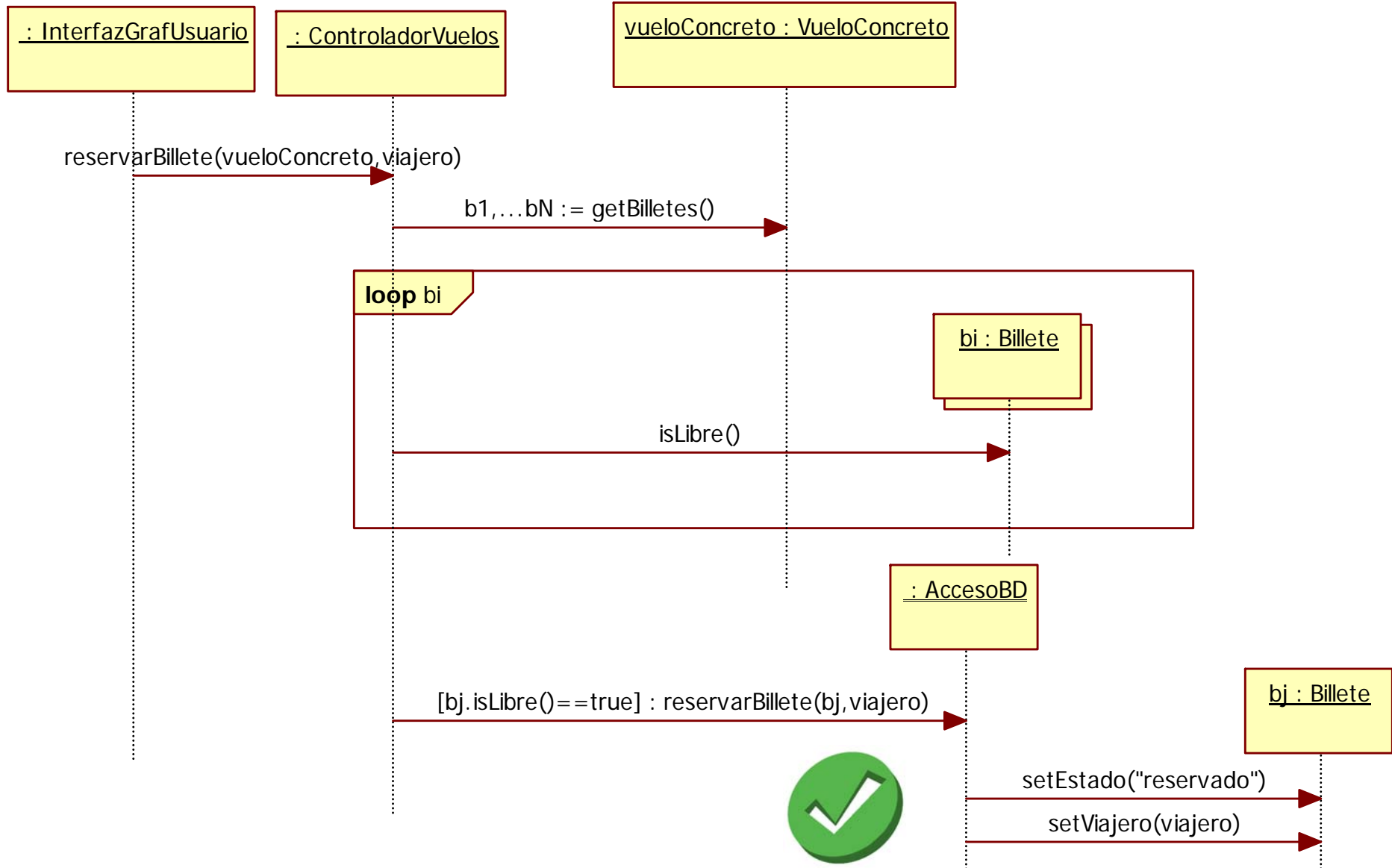


Error típico: realizar más acciones que la responsabilidad asignada

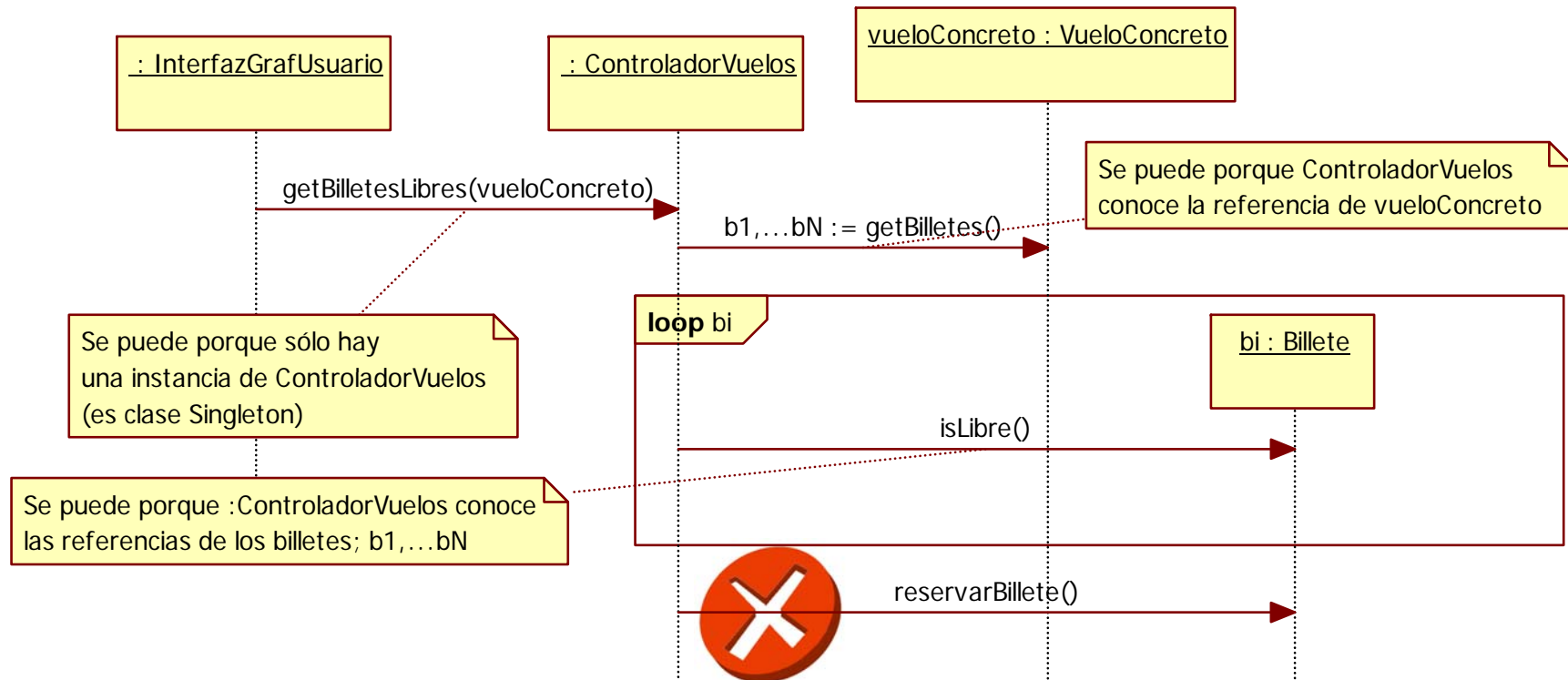


La responsabilidad era `getBilletesLibres(vueloConcreto)`
NO es coherente que dentro de ella se reserve un billete.
Nota: por otro lado, tampoco se haría así reservar billete (actualizando directamente un objeto del dominio)

Error típico: realizar más acciones que la responsabilidad asignada

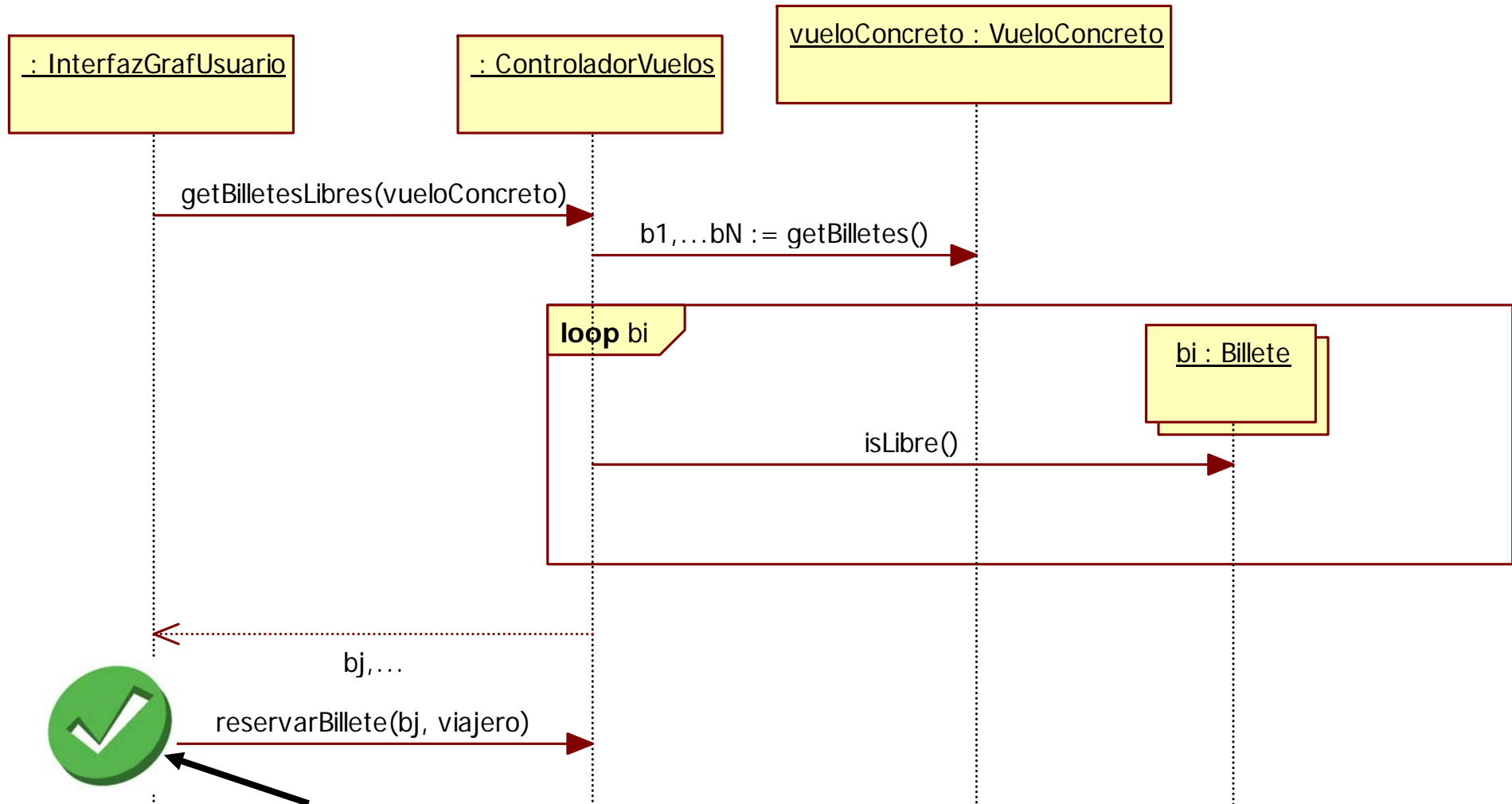


Error típico: no respetar el flujo de control en el diagrama de secuencia



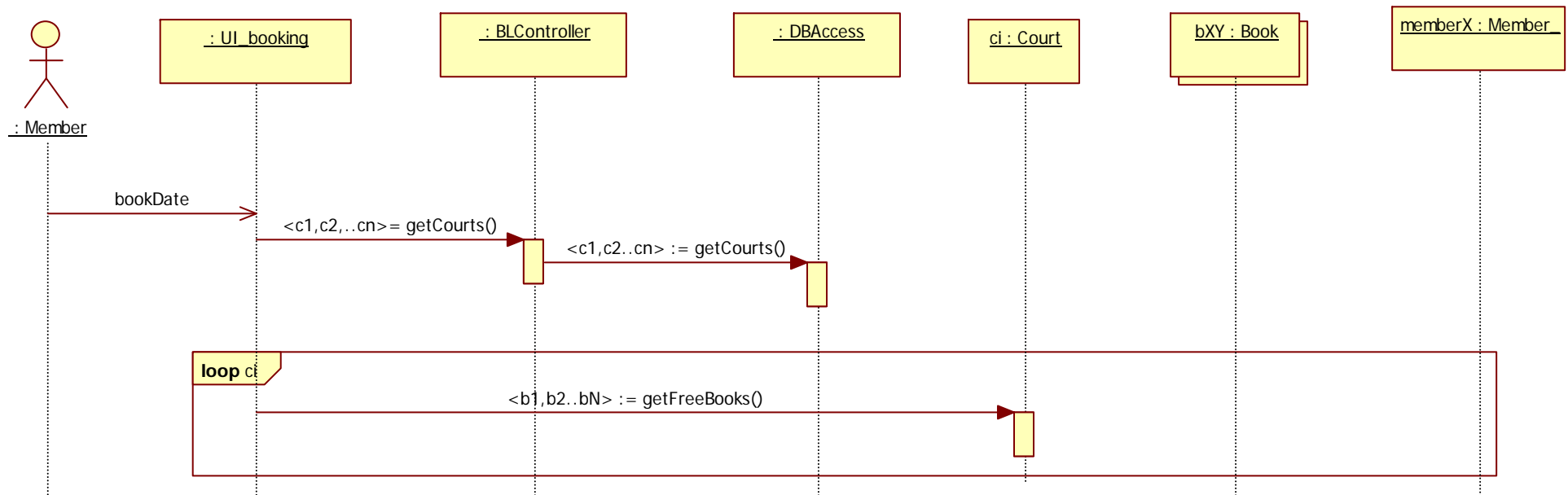
Tal vez la idea del diseño era, una vez obtenidos los billetes libres, entonces reservar un billete, pero en ese caso, el **FLUJO DE CONTROL** estaría en “InterfazGrafUsuario”, y **NO** en el objeto “ControladorVuelos”

Error típico: no respetar el flujo de control en el diagrama de secuencia



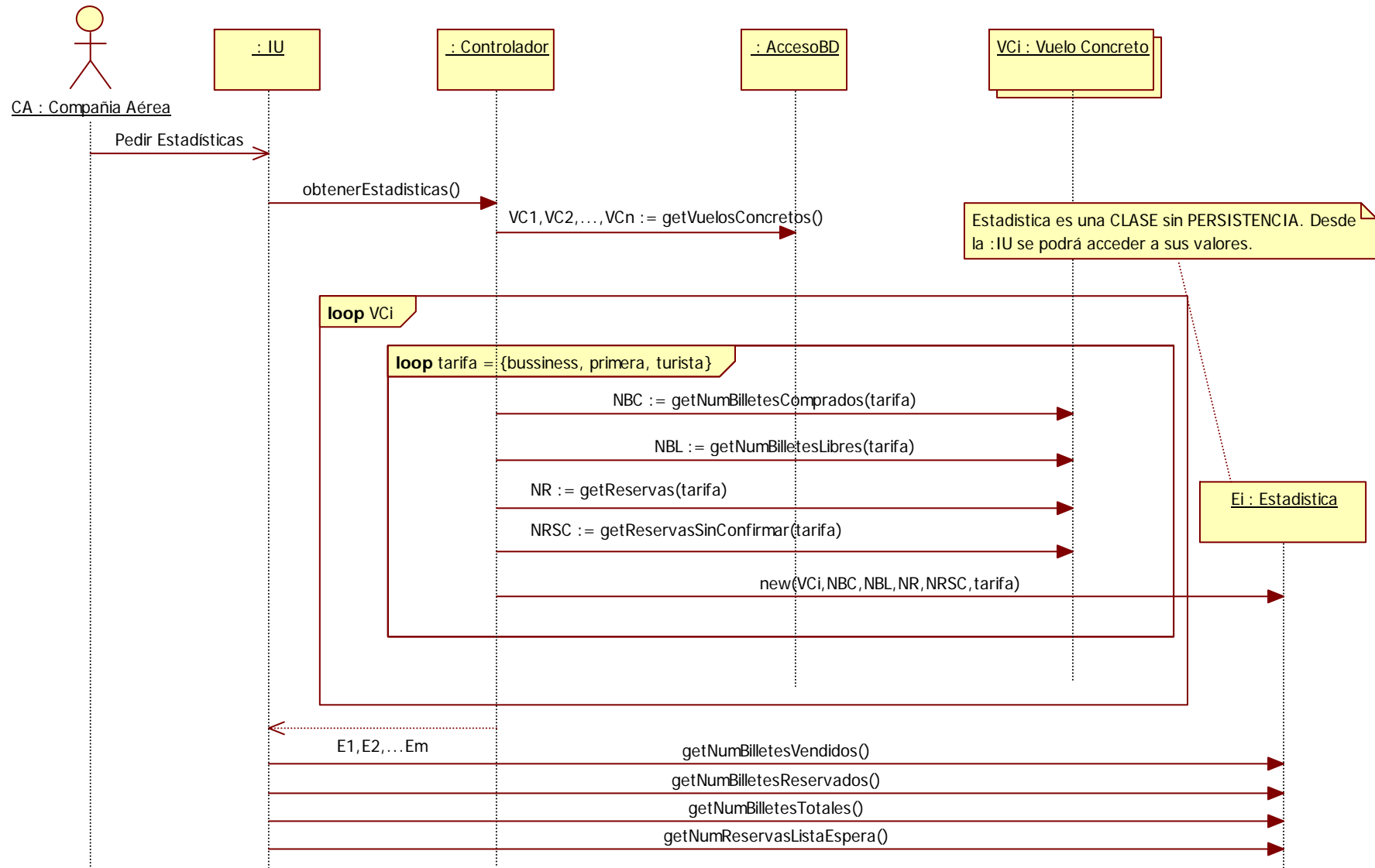
Aquí está el flujo de control, después de que “getBilletesLibres(vueloConcreto)” termine

Se puede ACCEDER a objetos del dominio directamente desde la interfaz, solamente para CONSULTAR datos



NOTA: en realidad, en la implementación veremos que se accede a COPIAS de dichos objetos del dominio y no a los objetos reales del dominio que realmente se encuentran en otra máquina remota (servidor de datos)

Se pueden CREAR objetos NO persistentes, [1) para enviar resultados a la interfaz]



[o 2) para diseñar algoritmos más complejos:

