

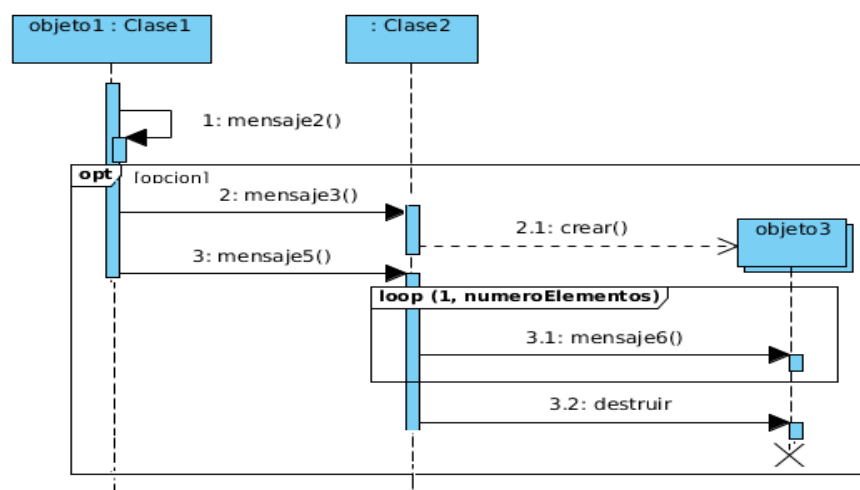
### Relación 3

**Ejercicio 1.** En general, ¿qué representan los diagramas de interacción de UML? ¿cuáles son sus componentes principales?

**Ejercicio 2.** Establece en una tabla la correspondencia entre los diagramas de secuencia y los diagramas de comunicación, indicando cómo se representan los siguientes elementos: objetos, mensajes, canal de comunicación, estructuras de control, subordinación en el envío de mensaje y orden de un mensaje en una secuencia de mensajes.

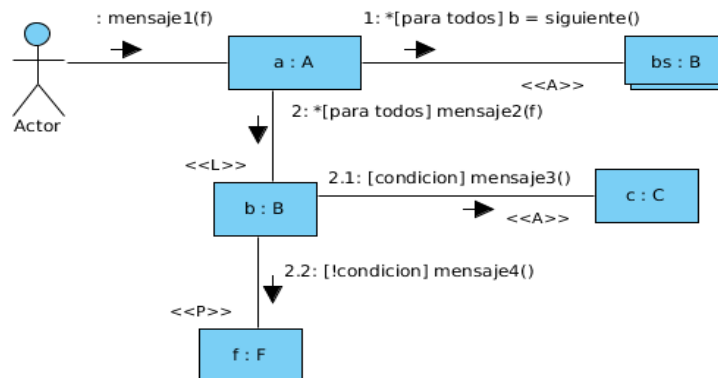
**Ejercicio 3.** ¿Qué relación existe entre el diagrama de clases y los diagramas de interacción?

**Ejercicio 4:** Dado el siguiente Diagrama de secuencia, responde verdadero (V) o falso (F) a las siguientes cuestiones



El envío de mensaje 1 ( <b>mensaje2()</b> ) es un envío de mensaje a <b>self</b> y además <b>recursivo</b>	
El objeto <b>objeto3</b> es un objeto que vive sólo en esta operación	
En la clase <b>Clase2</b> tienen que estar definidos los siguientes métodos: <b>mensaje3()</b> , <b>mensaje5()</b> y <b>mensaje6()</b>	
El <b>fragmento combinado</b> tipo <b>loop</b> solo se puede usar para el envío de mensajes a multiobjetos, tal y como está representado en el ejemplo	
La representación del multiobjeto está mal, falta especificar la clase a la que pertenecen los objetos que forman ese multiobjeto	
La numeración está mal, los envíos de mensaje 3.1 y 3.2 deberían ser 2.2 y 2.3	
El siguiente código Ruby <b>es correcto</b> : <pre> class Clase 2   def mensaje5     objeto3.each [ obj  obj.mensaje6()]     objeto3=nil   end end </pre>	

**Ejercicio 5:** Dado el siguiente Diagrama de comunicación, responde verdadero (V) o falso (F) a las siguientes cuestiones



El enlace o canal de comunicación estereotipado como **<<P>>** no puede ser de ese tipo ya que el objeto **f:F** no ha entrado como parámetro a la operación

La estructura de control usada en los envíos de mensajes números **2.1** y **2.2** es la estructura **if(condicion){...} else {...}**

En la clase B debe estar implementado el método **siguiente()** para poder responder al envío de mensaje número 1

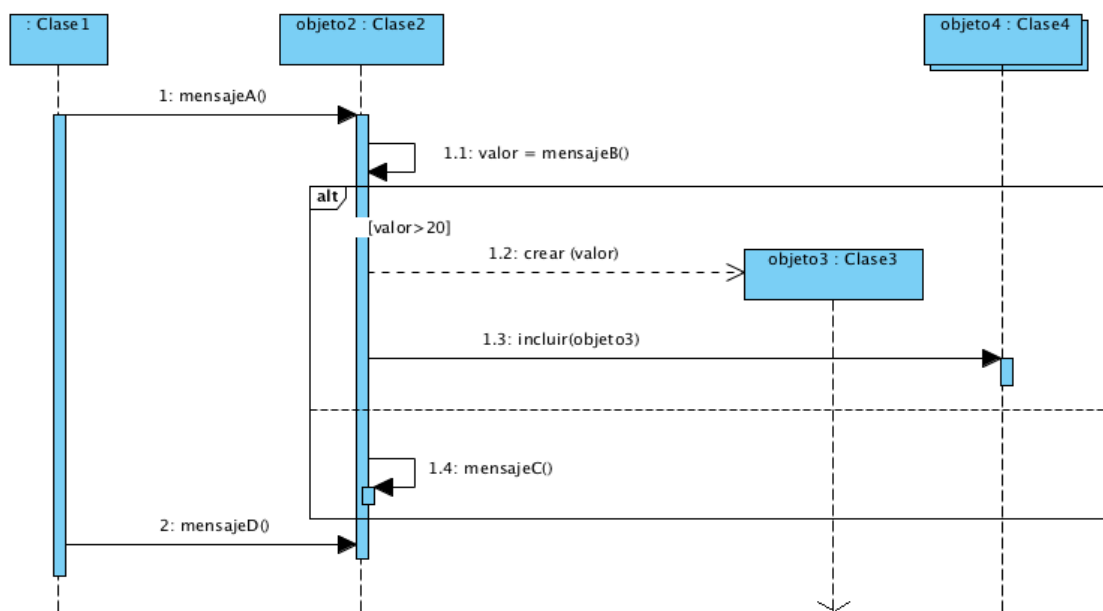
A los envíos de mensaje 2.1 y 2.2 les falta **\*[para todos]**

El siguiente código Java **es correcto**:

```

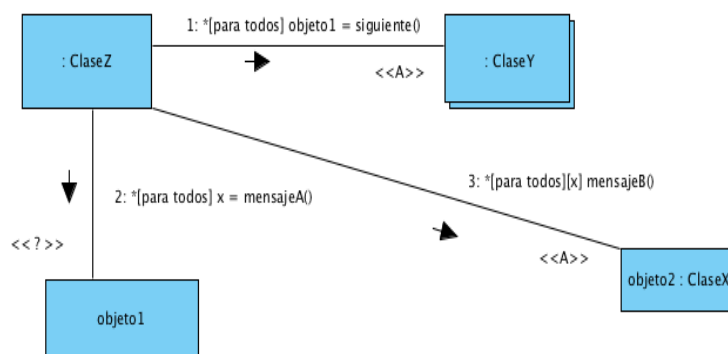
public class A {
    public void mensaje1(F f){
        for(B b:bs){
            b.mensaje2(f); } }
}
  
```

**Ejercicio 6.** A partir del siguiente diagrama de secuencia, responde verdadero (V) o falso (F) a las siguientes cuestiones



La clase <b>Clase2</b> debe tener implementado un método que se llame <b>mensajeB</b>	
El paso numerado con un 2 debería ser 1.5	
En el diagrama de comunicación equivalente, el tipo de enlace entre el <b>objeto2</b> y el <b>objeto3</b> sería <<A>> o <<L>>	
El paso 1.2 corresponde a una llamada al constructor por defecto de la clase <b>Clase3</b>	
El objeto <b>objeto4</b> es de la clase <b>Clase4</b>	
<b>mensajeB</b> y <b>mensajeC</b> se ejecutan de forma recursiva	
Los pasos 1.2, 1.3 y 1.4 se ejecutan cuando valor es mayor de 20	

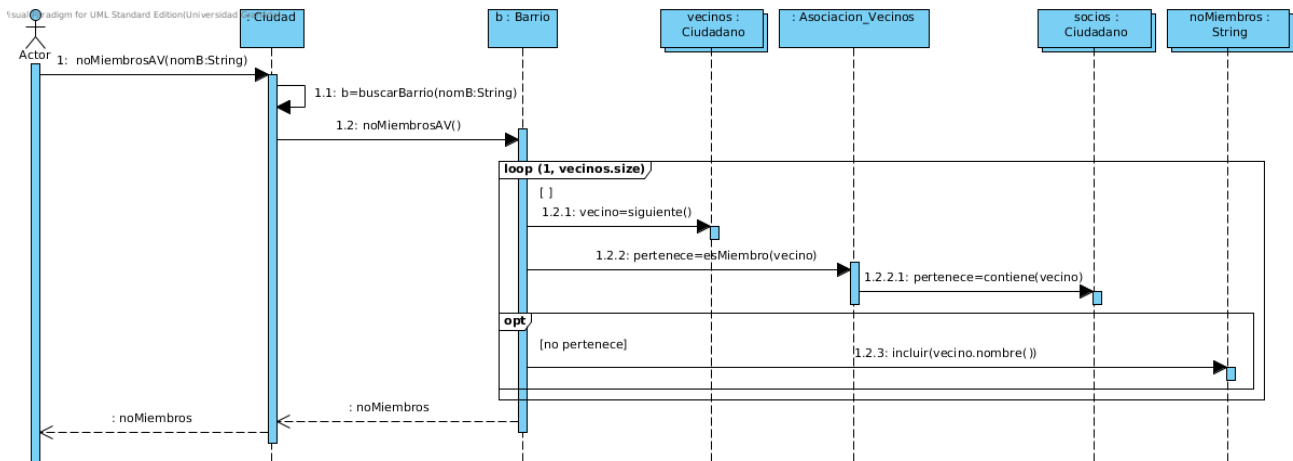
**Ejercicio 7.** A partir del siguiente diagrama de comunicación, responde verdadero (V) o falso (F) a las siguientes cuestiones



El tipo de enlace que aparece con interrogación no puede ser <<A>>	
Al codificar el diagrama, las instrucciones correspondientes a los pasos 1, 2 y 3 estarán todas dentro de un mismo bucle <b>for</b>	
Aunque no se indique explícitamente, es posible conocer la clase de <b>objeto1</b> por la información contenida en el diagrama	

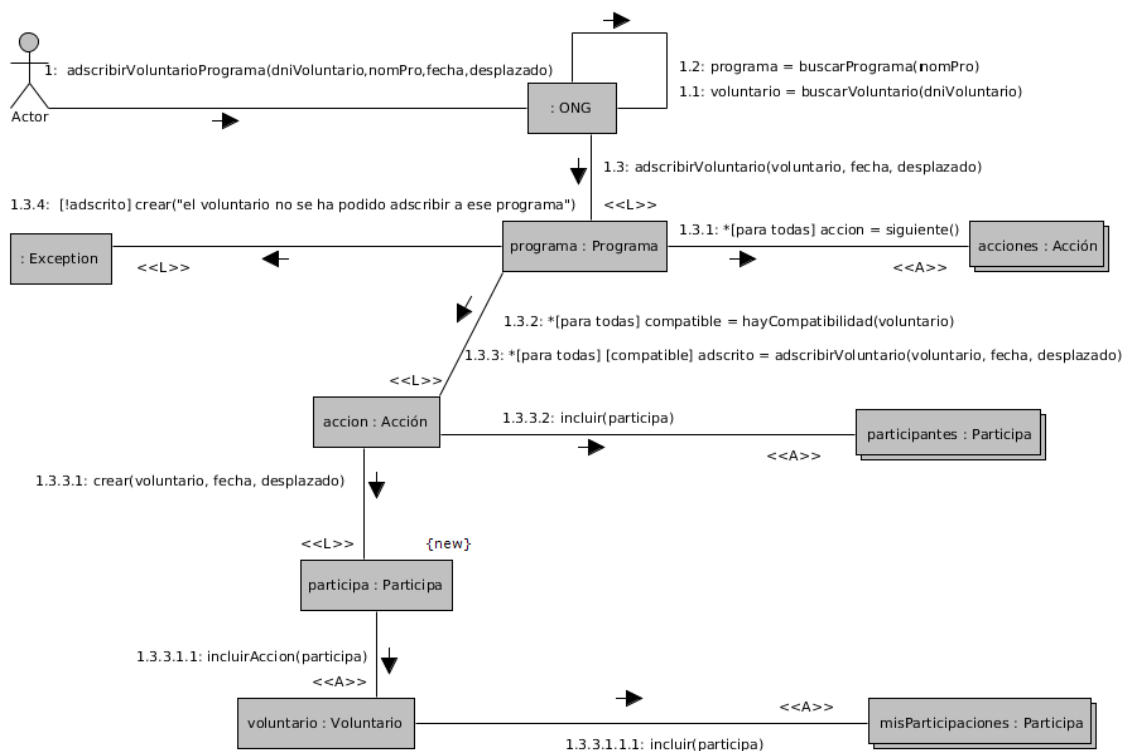
**Ejercicio 8.** Escribe el código correspondiente en Java y Ruby para los diagramas de los ejercicios 4 al 7.

**Ejercicio 9.** En relación al diagrama de clases del ejercicio 5 de la relación de ejercicios 2, tenemos el siguiente diagrama de secuencia para la operación **noMiembrosAv** de la clase Ciudad:



- Comprueba qué correspondencia hay entre el diagrama de clases y el de secuencia.
- Tradúcelo a Diagrama de comunicación
- Implementa en Java y en Ruby todos los métodos especificados en el diagrama de secuencia.

**Ejercicio 10.** A partir del siguiente diagrama de comunicación, correspondiente al ejercicio 6 de la relación 2, para la operación **AdscribirVoluntarioPrograma** de la clase ONG:

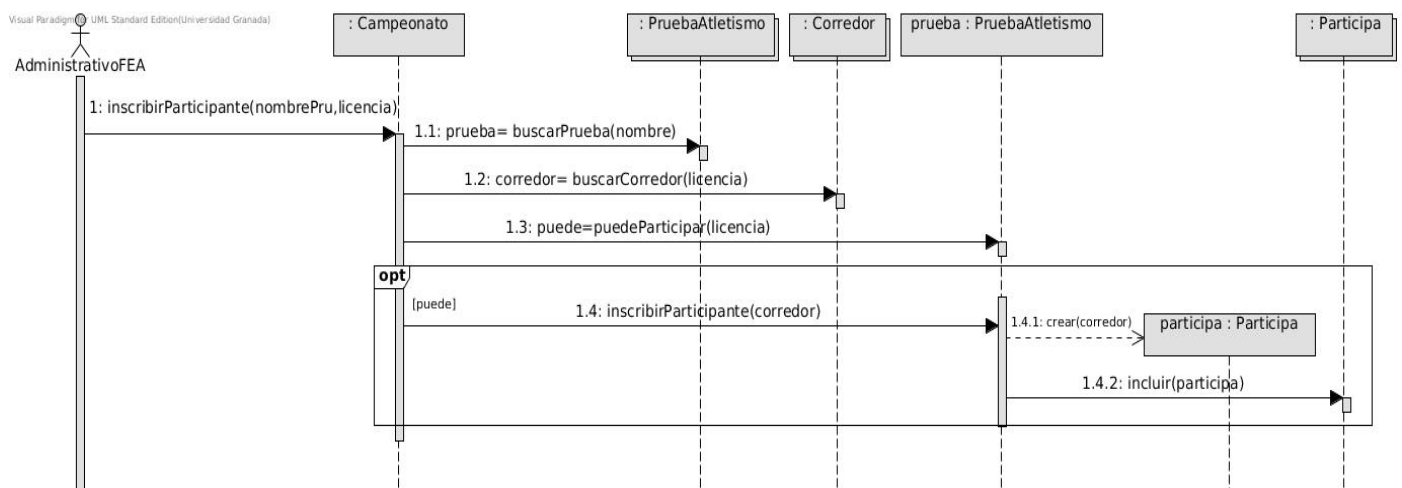


- A) Intenta entender qué se expresa en él y responde V(verdadero) o F(falso) a las siguientes cuestiones:

En el envío de mensaje 1.2 el objeto receptor es <b>self/this</b> .	
El envío de mensaje 1.3.1 significa que a todas las acciones del programa le vamos a adscribir un voluntario.	
En el método <b>crear</b> de la clase <b>Participa</b> (envío de mensaje 1.3.3.1 y subordinados) se construye un enlace entre el objeto <b>participa</b> y el objeto <b>voluntario</b> .	
El enlace entre el objeto <b>Acción</b> y el multiobjeto de la clase <b>Participa</b> estereotipado como <<A>> significa que el objeto <b>accion</b> conoce al multiobjeto sólo para esta operación.	
El multiobjeto <b>misParticipaciones</b> enlazado con <b>voluntario</b> es un subconjunto del multiobjeto <b>participantes</b> enlazado con <b>accion</b> .	
El envío de mensaje 1.3.3 se lleva a cabo sólo si <b>adscrito</b> es verdadero.	

- B) Implementa en Java y en Ruby el método **adscribirVoluntario(...)** de la clase Programa.
- C) Obtén el diagrama de secuencia de la operación **adscribirVoluntario(...)** de la clase **Accion**.
- D) Indica si en el diagrama de clases asociado falta o sobra alguna operación de las que aparecen en el diagrama de secuencia

### Ejercicio 11. A partir del siguiente diagrama de secuencia

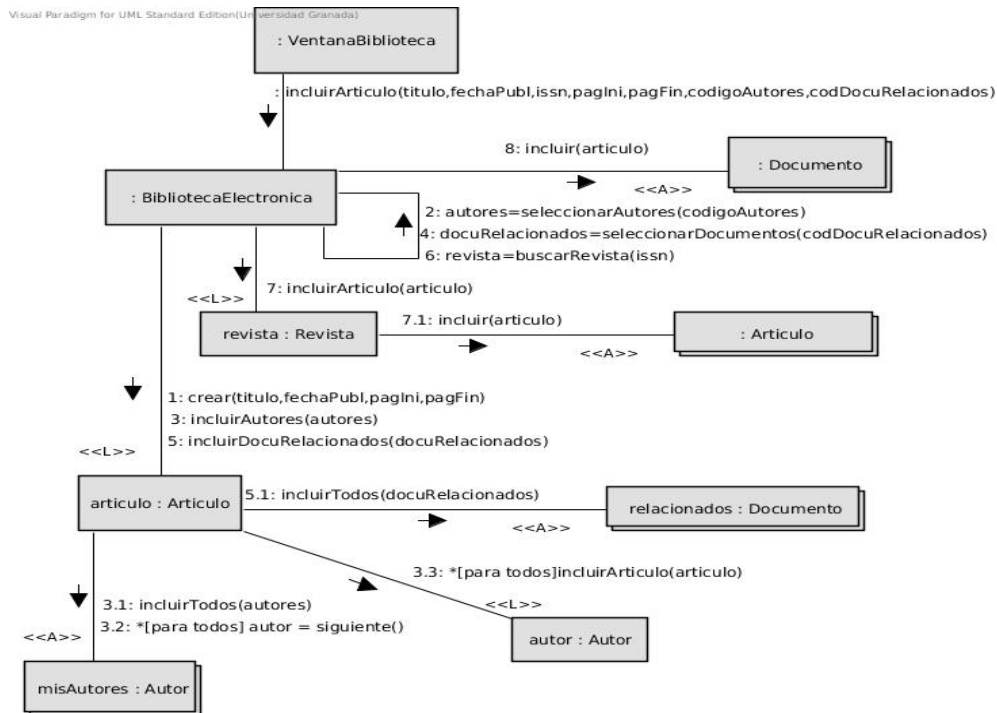


- A) Responde V (verdadero) o F (Falso) a las siguientes cuestiones:

La condición del fragmento combinado <b>opt</b> está mal, tendría que estar negada	
El envío de mensaje 1.4.2 está mal, el objeto receptor no es un multiobjeto	
El objeto <b>participa:Participa</b> se crea si (puede == true)	
El argumento del envío de mensaje 1.1 está mal debe ser <b>nombrePru</b>	
El envío de mensaje 1.4.1 se corresponde con la instanciación de un objeto de la clase <b>Participa</b>	
El objeto receptor del envío de mensaje 1.2 es un objeto de la clase <b>Corredor</b>	
Los envíos de mensaje 1.4.1 y 1.4.2 deberían estar fuera del fragmento combinado <b>opt</b>	

- B) Obtén el diagrama de comunicación equivalente
- C) Implementa en Java y el Ruby el método **inscribirParticipante** en la clase **Campeonato**

## Ejercicio 12. A partir del diagrama de comunicación:



A) Intenta comprenderlo y responde V(verdadero) y F (falso) a las siguientes cuestiones:

Los estereotipos de visibilidad no están especificados en el diagrama	
El objeto <b>relacionados:Documento</b> es un objeto de la clase <b>Documento</b>	
El envío de mensaje 5.1 está mal numerado debería ser 3.4	
En este diagrama se muestra que el objeto <b>revista:Revista</b> conoce a un multiobjeto de objetos de la clase <b>Artículo</b>	
La implementación del envío de mensaje 5.1 es: relacionados.incluirTodos(docuRelacionados)	
La implementación del método <b>incluirAutores(autores)</b> en la clase <b>Articulo</b> es la siguiente y es correcta: <pre> class Articulo     private ArrayList&lt;Autor&gt; autores = new ArrayList();     void incluirAutores(Autor autores){         misAutores.addAll(autores);         for(Autor autor:misAutores)             autor.incluirArticulo(articulo);     } } </pre>	
Los envíos de mensaje 2, 4 y 6 son envíos de mensajes a self	

- B) Implementa en Java y el Ruby el método **incluirArticulo(...)** en la clase **BibliotecaElectronica**
- C) Obtén el Diagrama de secuencia de la operación **3:incluirAutores(autores)**

**Ejercicio 13.** Partiendo de los diagramas de clases obtenidos en la relación de problemas de la relación 2 en el ejercicio 10 de los supuestos B y C, obtener el diagrama de interacción (secuencia o comunicación) de las siguientes operaciones. Se pueden añadir atributos o modificar el diagrama de clases si se considera necesario.

- A) (del 10.B) Obtener los resultados de la carrera celebrada en una determinada fecha y lugar, se debe proporcionar el resultado por equipos e individual por atletas (medallas de oro, plata y bronce). Como salida se debe proporcionar fecha, lugar y categoría de la carrera. para los equipos ganadores, indicar el nombre del equipo, el tiempo invertido y el nombre de los atletas que lo componen y para los resultados individuales, el nombre del atleta y el tiempo invertido.
- B) (del 10.C) Matricular a un alumno de una asignatura en un grupo concreto, la asignatura es identificada por un código y el grupo por un letra que es su denominación. Terminada la operación se ha enlazado un objeto alumno con el grupo de una asignatura.