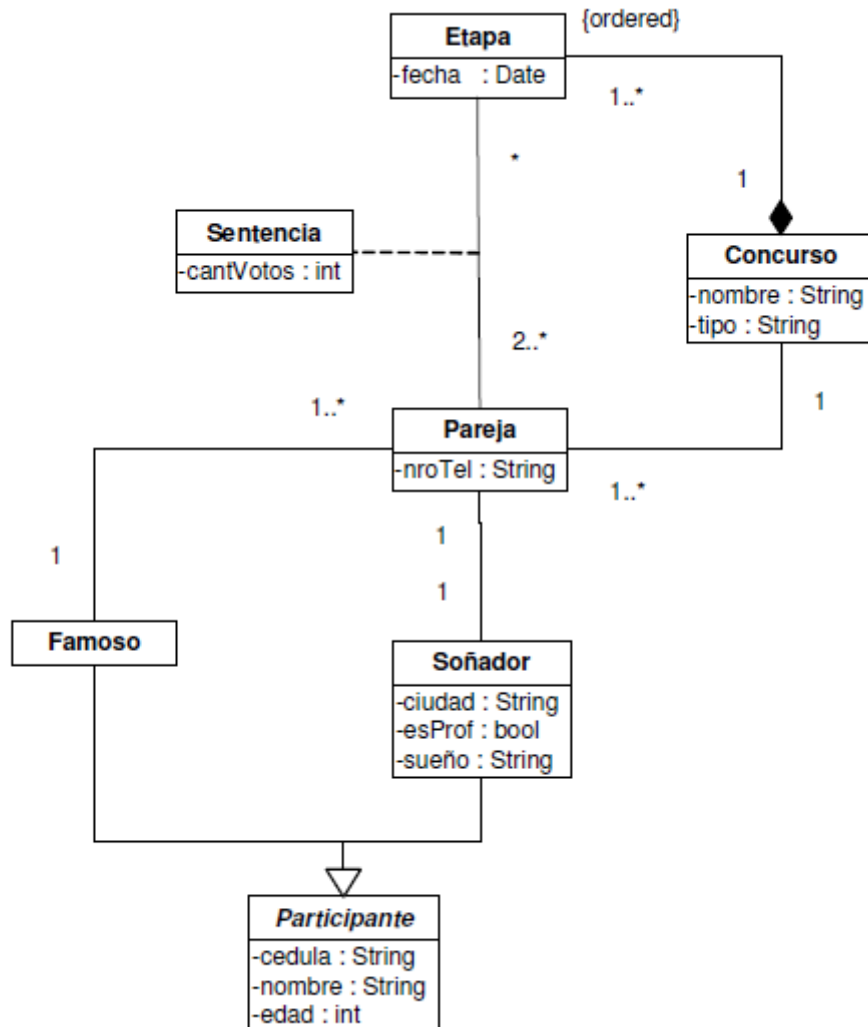


## Solución Primer Parcial de programación avanzada – 2011

Ejercicio 1 – Ver diapositivas de teórico

Ejercicio 2.



Restricciones:

context Etapa inv:

-- En una etapa, las parejas sentenciadas deben competir en el concurso al que la etapa pertenece.

`self.pareja->forall(p | p.concurso = self.concurso)`

-- La pareja con menor cantidad de votos en una etapa no participa de una sentencia posterior

let eliminada : Pareja = self.sentencia->select(s1 | self.sentencia->forall(s2 | s2 <> s1 implies s1.cantVotos < s2.cantVotos) )->any().pareja in eliminada.etapa->forall(e | e.fecha <= self.fecha)

context Famoso inv:

-- Un famoso no puede formar parte de más de una pareja en un mismo concurso.

self.pareja->forAll(p1, p2 | p1 <> p2 implies p1.concurso <>p2.concurso)

context Concurso inv:

-- Un concurso se identifica por su nombre y tipo.

Concurso.allInstances()->forAll(c1, c2 | c1 <> c2 implies c1.nombre <> c2.nombre or c1.tipo <> c2.tipo)

-- Para un concurso no puede haber dos etapas con una misma fecha.

self.etapa->isUnique(fecha)

-- Para un concurso no puede haber dos parejas con el mismo número de teléfono.

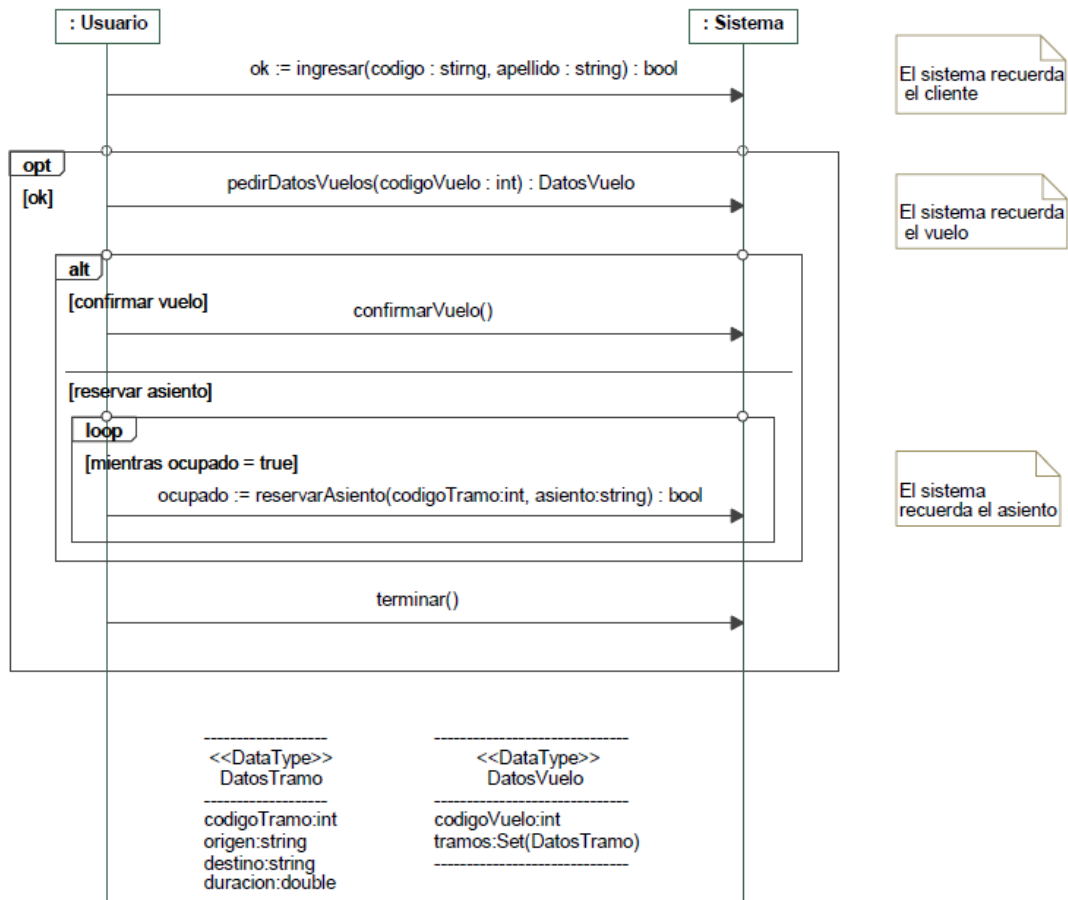
self.pareja->isUnique(nroTel)

context Participante inv:

-- Un participante se identifica por su cédula.

Participante.allInstances()->isUnique(cedula)

### Ejercicio 3



#### Ingresar

Pre:

Post: El sistema devuelve verdadero si existe un usuario de apellido= datavalue apellido y existe una reserva de código = datavalue código asociado al usuario, el sistema no cambia.

#### pedirDatosVuelos

Pre: existe un Vuelo en el sistema con atributo código = codigoVuelo

Post: El sistema devuelve un Datavalue de tipo DatosVuelo, con códigoVuelo = código del vuelo buscado, y tramos = Set de Datatramos para ese vuelo con codigoTramo, origen, destino y duración iguales a los tramos del vuelo buscado. No hay cambios en el sistema, ni nuevas instancias.

#### ConfirmarVuelo

Pre:

Post: Se guarda en memoria del sistema la opción de confirmar el vuelo, el sistema no cambia

#### reservarAsiento

Pre: existe un tramo en el sistema con código = codigoTramo

Post: El sistema devuelve verdadero si para el tramo de código=codigoTramo el asiento esta libre, y falso en caso contrario, el sistema recuerda en memoria el asiento, el sistema no cambia, no hay nuevas instancias ni links.

Terminar:

Pre:

Post: Se creó un nuevo link entre vuelo de códigoVuelo y usuario de apellido = datavalue apellido.

Se creó un link entre usuario de apellido = datavalue apellido y tramo de codigoTramo

Se agregó un asiento con valor datavalue asiento a la colección de asientos del tramo anterior.