

Programación 4

PRÁCTICO 4

Los ejercicios marcados con * se consideran indispensables para adquirir los conocimientos básicos del curso.

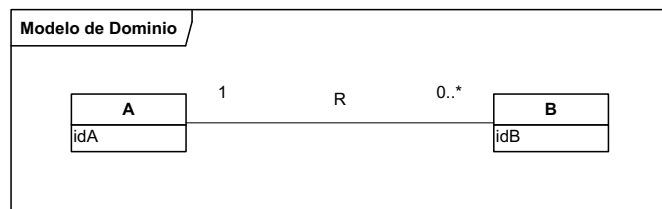
Ejercicio 1 *

Construir los diagramas de comunicación que reflejen las siguientes interacciones:

- Un objeto de clase *ClaseA* recibe como punto de entrada un mensaje *mensaje1()* y si la condición *cond* se satisface, envía un mensaje *mensaje2()* a un objeto de clase *ClaseB*, y en caso contrario, envía un mensaje *mensaje3()* a un objeto de clase *ClaseC*.
- Un objeto de clase *ClaseA*, al recibir el mensaje *mensaje1()*, crea una instancia de clase *ClaseB*, y se lo envía a un objeto de clase *ClaseC*, el cual envía el mensaje *mensaje3()* a la instancia recién creada.
- Un objeto de clase *ClaseA*, al recibir el mensaje *mensaje1()*, envía *m* veces el mensaje *mensaje2()* a una instancia de clase *ClaseB*, que a su vez, por cada mensaje recibido, envía un mensaje *mensaje3()* a un objeto de clase *ClaseC* y otro mensaje *mensaje4()*, a uno de clase *ClaseD*.
- Un objeto de clase *ClaseA* recibe como punto de entrada un mensaje *mensaje1()*, y envía a cada objeto de una colección de objetos de clase *ClaseB* el mensaje *mensaje2()*.

Ejercicio 2 *

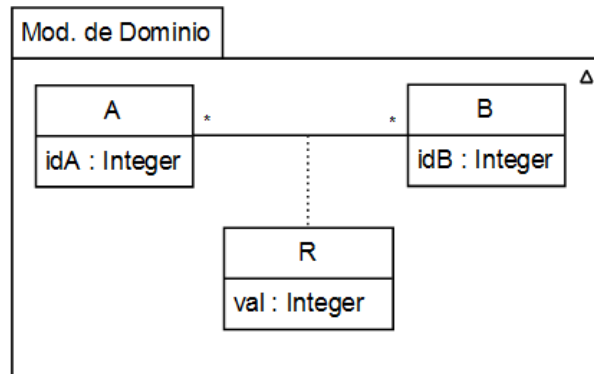
Considerar el Modelo de Dominio presentado en el siguiente diagrama.



- Realizar el diagrama de comunicación en el que una instancia de A crea una de B con link unidireccional de A hacia B.
- Realizar el diagrama de comunicación en el que una instancia de B crea una de A con link unidireccional de B hacia A.
- Modificar los diagramas anteriores de forma que el link creado sea bidireccional.
- Discutir la destrucción de instancias y links en los diferentes escenarios anteriores.

Ejercicio 3 *

Considerar el Modelo de Dominio presentado en el siguiente diagrama.



Se desea diseñar la operación que retorne todos los identificadores de las instancias de *A* que participen en la asociación *R* cuyo atributo *val* sea mayor que un valor dado.

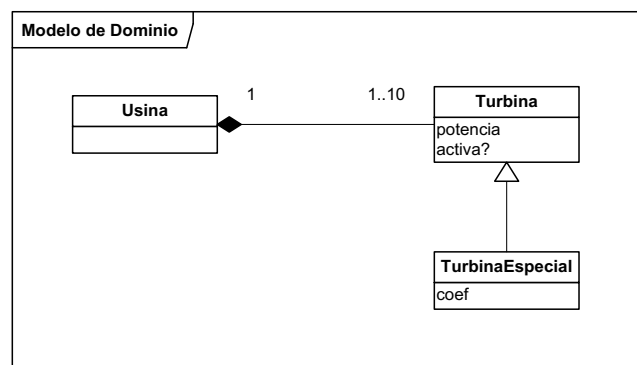
- Realizar el diagrama de comunicación sabiendo que *R* es navegable de *A* hacia *B*.
- Realizar el diagrama de comunicación sabiendo que *R* es navegable de *B* hacia *A*.

Ejercicio 4

Considerar el sistema analizado en el Ejercicio 3 del Práctico 3 referente a la venta de vehículos a personas. Diseñar la interacción de la colaboración que realiza el caso de uso que registra la venta de un vehículo a una persona.

Ejercicio 5 *

Considerar el Modelo de Dominio presentado en el siguiente diagrama. El atributo *potencia* indica la capacidad de generación de energía de una turbina, que en el caso de ser especial se multiplica por el valor del atributo *coef*.



Diseñar la interacción de la colaboración que realiza el caso de uso Obtener Potencia de una Usina que obtiene la potencia total que son capaces de generar todas las turbinas activas de una usina.

Ejercicio 6

Considerar el siguiente documento de Visión del Problema.

El Ministerio de Salud Pública ha decidido verificar la calidad de los medicamentos producidos por ciertas empresas del sector. A tales efectos se ha dispuesto que sus inspectores estudien la composición de los medicamentos producidos por estas empresas (realizando un análisis químico en un laboratorio del MSP) a fin de comprobar que la misma coincida con la especificada para cada medicamento.

Se desea diseñar una operación mediante la cual el jefe de la investigación le pide a un inspector que inspeccione algunas de las empresas, verificando la calidad de una muestra de los medicamentos producidos por esa empresa. El inspector debe entregar un informe en el que conste, para cada empresa inspeccionada, el nombre de todos los medicamentos verificados junto con el porcentaje de medicamentos defectuosos de cada tipo.

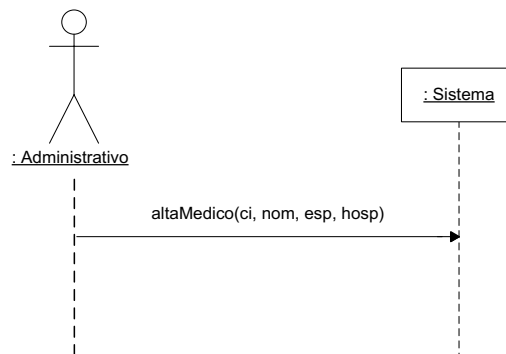
El informe debe permitir llenar fácilmente el siguiente formulario:

Nombre Empresa A	
Nombre Medicamento A	Porcentaje defectuoso
Nombre Medicamento B	Porcentaje defectuoso
...	
Nombre Empresa B	
...	

- Representar la operación indicada mediante diagramas de comunicación en UML de manera que refleje fielmente la descripción dada, o sea, tal como ocurre en la realidad sin tener en cuenta consideraciones de diseño.
- Evalúe el diseño anterior en función de los criterios GRASP.
- Diseñe la operación apropiadamente de manera de solucionar los posibles inconvenientes detectados. Justifique el nuevo diseño indicando los GRASP utilizados, indicando en qué aspectos el nuevo diseño es superior al anterior.

Ejercicio 7

Considerar la realidad del Ejercicio 2 del Práctico 2 referente a hospitales, médicos y pacientes. Para el caso de uso Alta Médico se ha realizado el siguiente DSS y el siguiente contrato de software para la operación de sistema identificada.

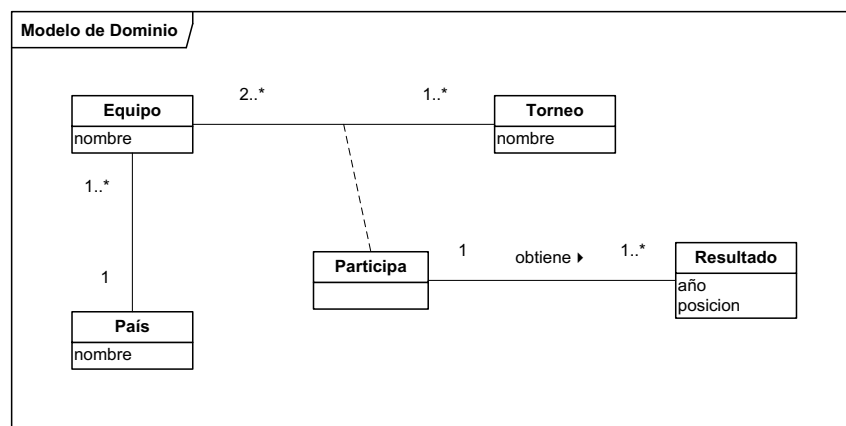


Nombre	Alta Médico
Operación	<code>altaMedico(ci:String, nom:String, esp:String, hosp:String)</code>
Entrada	<i>ci</i> la cédula del médico, <i>nom</i> el nombre del médico, <i>esp</i> la especialidad del médico, <i>hosp</i> identificador del hospital en el que trabaja el médico.
Descripción	Dar de alta el médico en el sistema registrando sus datos y el hospital en el que trabaja.
Pre- y poscondiciones	
pre1: No hay médicos registrados con la cédula <i>ci</i> .	
pre2: El hospital identificado por <i>hosp</i> está registrado en el sistema.	
post: Existe una nueva instancia de Médico con los datos <i>ci</i> , <i>nom</i> y <i>esp</i> , y existe un link entre este médico y el hospital identificado por <i>hosp</i> .	

Diseñar la interacción correspondiente a la colaboración que realiza el caso de uso Alta Médico.

Ejercicio 8 *

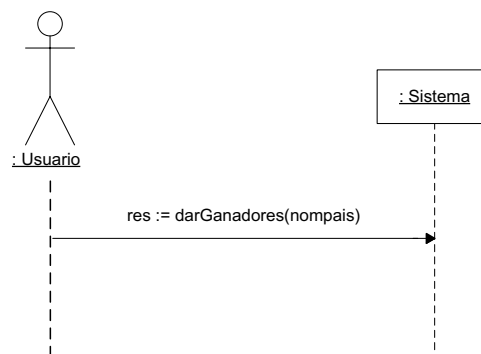
Considerar el siguiente Modelo de Dominio.



Diseñar las interacciones para las colaboraciones que realizan los siguientes casos de uso.

a) "Dado un país, indicar todos los equipos que han ganado algún torneo."

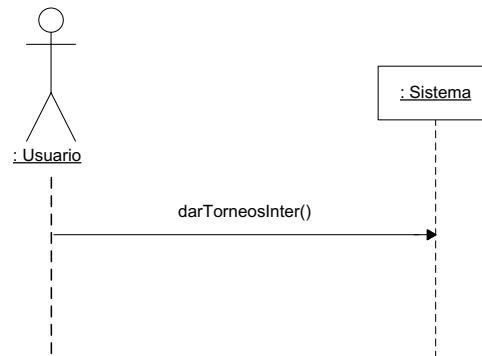
Para este caso de uso, se realizaron los siguientes artefactos durante el análisis.



Nombre	Dar Ganadores
Operación	<code>darGanadores(nompais:String):Set(String)</code>
Entrada	Nombre del país.
Salida	El conjunto de nombres de equipos.
Descripción	Dado el nombre de un país, devuelve el conjunto de nombres de equipos que han ganado algún torneo
Pre- y poscondiciones	
pre: Existe el país con el nombre indicado por <code>nompais</code> .	
post: El conjunto resultado equivale a los nombres de todos los equipos del país identificado por <code>nompais</code> que han ganado algún torneo, considerando que un equipo ha ganado un torneo si habiendo participado en él obtuvo la posición 1.	

b) “Dar los nombres de todos los torneos internacionales.”

Para este caso de uso, se realizaron los siguientes artefactos durante el análisis.

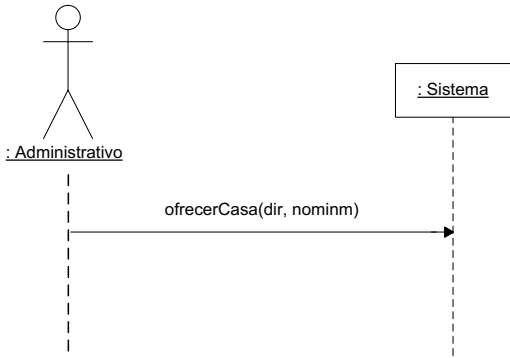


Nombre	Dar Torneos Internacionales
Operación	<code>darTorneosInter():Set(String)</code>
Salida	El conjunto de nombres de torneos internacionales.
Descripción	Devuelve el conjunto de nombres de torneos internacionales.
Pre- y poscondiciones	
post: El conjunto resultado equivale a los nombres de todos los torneos tales que no todos los equipos que participan en ellos pertenecen al mismo país.	

Ejercicio 9

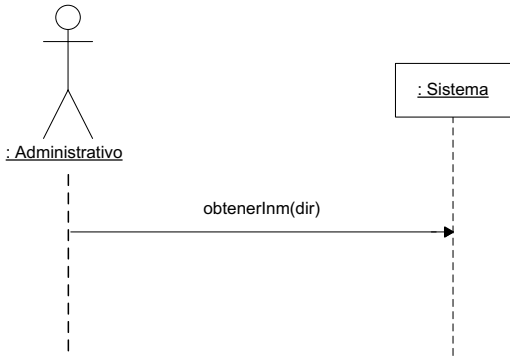
Considerar la realidad del Ejercicio 9 del Práctico 2 referente a negocios inmobiliarios y los artefactos de análisis correspondientes a los casos de uso Ofrecer Casa, Obtener Inmobiliarias y Agendar Visita. Diseñar las interacciones para las colaboraciones que realizan estos casos de uso.

a) Caso de uso Ofrecer Casa



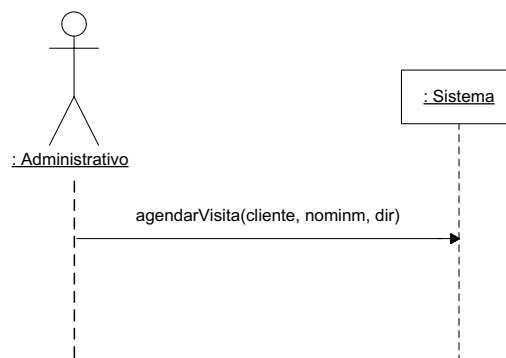
Nombre	Ofrecer Casa
Operación	<code>ofrecerCasa(dir:String, nominm:String)</code>
Entrada	<i>dir</i> es la dirección de la casa que se desea ofrecer, <i>nominm</i> es el nombre de la inmobiliaria que ofrecerá la casa.
Descripción	Registrar que la casa ubicada en <i>dir</i> sea ofrecida por la inmobiliaria <i>nominm</i> .
Pre- y poscondiciones	
pre1: La casa dada por <i>dir</i> existe en el sistema.	
pre2: La inmobiliaria dada por <i>nominm</i> existe en el sistema.	
post: Existe un link entre la casa identificada por <i>dir</i> y la inmobiliaria identificada por <i>nominm</i> .	

b) Caso de uso Obtener Inmobiliarias



Nombre	Obtener Inmobiliarias
Operación	<code>obtenerInm(dir:String) :Set (String)</code>
Entrada	<i>dir</i> es la dirección de la casa.
Salida	El conjunto de nombres de inmobiliarias que ofrecen la casa.
Descripción	Devuelve el conjunto de nombres de todas las inmobiliarias que ofrecen la casa.
Pre- y poscondiciones	
pre: La casa dada por <i>dir</i> existe en el sistema.	
post: El conjunto devuelto equivale a los nombres de todas las inmobiliarias que ofrecen la casa.	

c) Caso de uso Agendar Visita



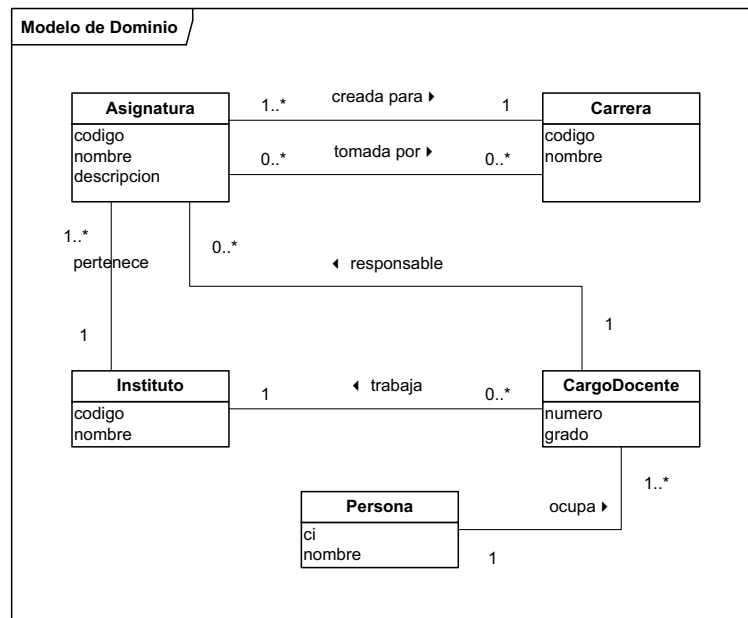
Nombre	Agendar Visita
Operación	<code>agendarVisita(cliente:String, nominm:String, dir:String)</code>
Entrada	<i>cliente</i> es el nombre del cliente, <i>nominm</i> es el nombre de la inmobiliaria y <i>dir</i> es la dirección de la casa.
Descripción	Agendar la visita del cliente a la casa a través de la inmobiliaria.
Pre- y poscondiciones	
pre1: La casa dada por <i>dir</i> existe en el sistema.	
pre2: La inmobiliaria identificada por <i>nominm</i> existe en el sistema.	
pre3: El cliente identificado por <i>cliente</i> existe en el sistema.	
post: Se registró que el cliente visitará la casa a través de la inmobiliaria.	

Ejercicio 10

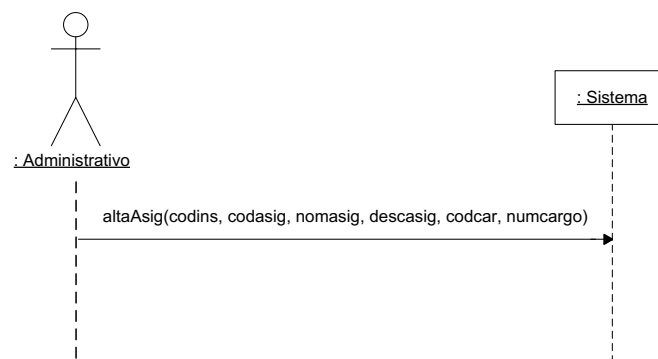
A partir del siguiente documento de Visión del Problema se construyó el Modelo de Dominio correspondiente.

La Facultad de Ingeniería está organizada en Institutos. Cada Instituto cuenta con un plantel docente que realiza, entre otras, tareas de enseñanza. Las asignaturas que se dictan en la facultad son responsabilidad de los institutos; cada asignatura pertenece a un único instituto, éste es el responsable de crear las asignaturas a su cargo, generando su contenido y demás detalles, asociándole un docente como responsable desde ese momento. Las asignaturas son creadas para su uso por una carrera determinada de la facultad; después que existen, sin embargo, pueden ser tomadas por otras carreras. A la facultad no le interesa mantener información de las asignaturas, cuando la necesita, se la pide al instituto que corresponda.

Las personas que trabajan como docentes pueden hacerlo en varios institutos (por lo menos en uno). Estas personas ocupan cargos docentes en cada instituto. Los cargos son propiedad de los institutos, o sea que un cargo docente pertenece a un único instituto. De los cargos docentes se conoce el número de cargo y el grado.



Para el caso de uso Alta Asignatura se realizaron los siguientes artefactos de análisis.



Nombre	Alta Asignatura
Operación	<code>altaAsig(codins:String, codasig:String, nomasig:String, descasig:String, codcar:String, numcargo:Integer)</code>
Entrada	<code>codins</code> es el código del instituto, <code>codasig</code> es el código de la asignatura, <code>nomasig</code> es el nombre de la asignatura, <code>descasig</code> es la descripción de la asignatura, <code>codcar</code> es el código de la carrera y <code>numcargo</code> es el número del cargo docente responsable.
Descripción	Da de alta una asignatura en el sistema
Pre- y poscondiciones	
<pre>pre1: El instituto <code>codins</code> existe en el sistema. pre2: El cargo docente <code>numcargo</code> existe en el sistema. pre3: La carrera <code>codcar</code> existe en el sistema. pre4: No existe en el sistema una asignatura con código <code>codasig</code>. post1: Existe una nueva instancia de Asignatura en el sistema cuyos atributos llevan los valores <code>codasig</code>, <code>nomasig</code> y <code>descasig</code>. post2: La nueva asignatura y el instituto <code>codins</code> están linkeados. post3: La nueva asignatura y la carrera <code>codcar</code> están linkeados por la asociación <i>creada para</i>. post4: La nueva asignatura y el cargo docente <code>numcargo</code> están linkeados.</pre>	

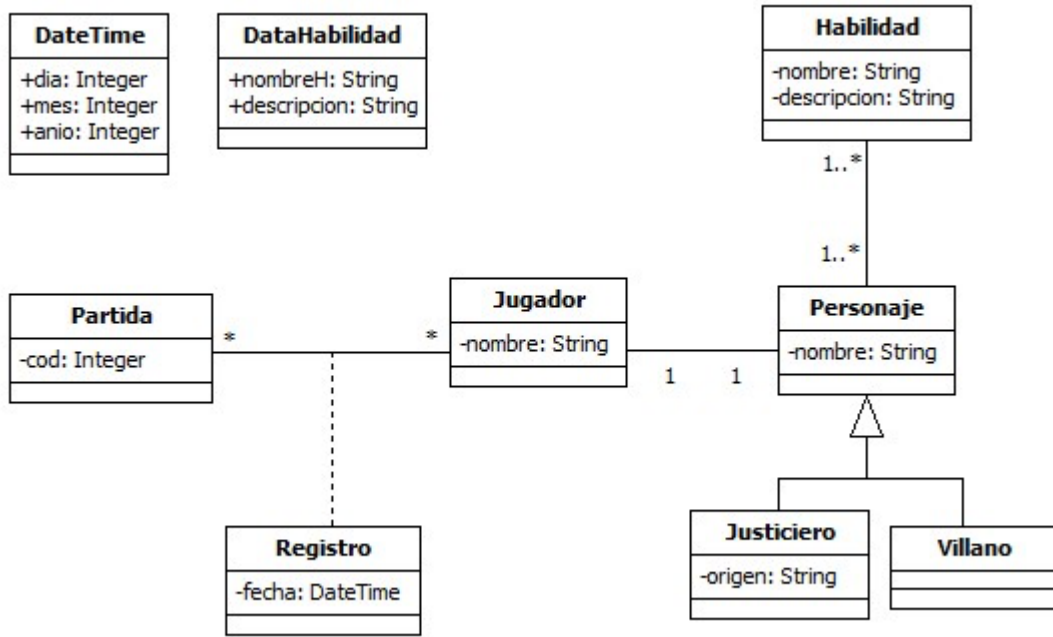
Diseñar las interacciones para la colaboración que realiza este caso de uso. Indicar en el diagrama los criterios de asignación de responsabilidades utilizados.

Ejercicio 11 (Examen Julio 2013)

A usted y a su grupo de desarrollo de software se le ha encargado la construcción de un pequeño Sistema, que tiene la finalidad de llevar el registro de las distintas partidas de un juego específico. En estas partidas de juego (identificadas por un código) participan jugadores (identificados por su nombre) asociados a su personaje, interesando saber la fecha en que estos jugadores se registran como participantes de esas partidas.

Los personajes poseen un conjunto de habilidades que irán adquiriendo en el desarrollo del juego. Estos pueden ser justicieros o villanos, interesando en los justicieros su origen. Los Personajes y las Habilidades se identifican también por su nombre.

Su equipo de análisis ha construido un modelo de dominio para el problema (ver siguiente página). Además, se consideran los casos de uso “Crear Jugador” y “Obtener información de un Jugador”, cada uno de los cuales es modelado con una sola operación del sistema, cuyos contratos se especifican a continuación (ver siguiente página).



crearNuevoJugadorEnPartida(codP: Integer, nomJ, nomP, tipo, origen: String, listdh: set(DataHabilidad), fActual: DateTime)	
Descripción	Ingresa un nuevo jugador con su personaje a una partida dada.
Parámetros	<ul style="list-style-type: none"> - codP: código de la Partida. - nomJ: nombre del nuevo Jugador. - nomP: nombre del Personaje. - tipo: puede tomar los valores "villano" o "justiciero". - origen: aplica si tipo="justiciero". - ldh: conjunto de habilidades iniciales del Personaje. - fActual: fecha actual del Sistema.
Precondiciones	<ul style="list-style-type: none"> - No existe en el sistema una instancia de Jugador con nombre igual a nomJ. - No existe en sistema una instancia de Personaje con nombre igual a nomP. - No existen en el sistema instancias de Habilidad con nombre igual a los atributos nombreH que vienen como parámetros en listdh. - Existe en el Sistema una instancia de PartidaDelJuego con código a cod.
Postcondiciones	<ul style="list-style-type: none"> - Existe en el Sistema una nueva instancia de Jugador cuyo atributo nombre es nomJ. - Se crea una instancia de tipo asociativo de Registro con fecha igual a fActual entre la Partida y el nuevo Jugador. - Existe en el Sistema un Personaje Villano o Justiciero. dependiendo de tipo, con los atributos nomP y origen si aplica. - Por cada data value dh en listdh, existe en el sistema una nueva instancia de Habilidad cuyos atributos se definen con los valores que tiene dh y se agrega a la colección de habilidades del Personaje. - Se crea un link entre el nuevo Jugador y su Personaje.

obtenerInfoJugador(nomJ : String) : DataJugador

Descripción	Devuelve información de un jugador
Parámetros	- nomJ: nombre del jugador
Precondiciones	- Existe en el sistema una instancia de Jugador identificada por nomJ.
Postcondiciones	<p>Se retorna un datavalue con la siguiente información:</p> <ul style="list-style-type: none"> - Nombre del Jugador cuyo nombre es igual a nomJ. - Todos los códigos de las Partidas en las que participó incluyendo su fecha de registro en cada Partida. - El nombre de su Personaje asociado, incluyendo su origen si es Justiciero. <p>(*) Obs: No es necesario mostrar las habilidades del Personaje</p>

Se pide:

Realizar los Diagramas de Comunicación de las dos operaciones descritas, indicando el tipo de visibilidad en todos los mensajes y definiendo los nuevos DataTypes que puedan haber surgido, incluyendo DataJugador.

Ejercicio 12 * (Laboratorio 2008)

Considerando la realidad presentada en prácticos anteriores, en esta etapa se debe diseñar las operaciones del sistema detectadas en el Ejercicio 8 del Práctico 3 para los siguientes casos de uso:

- Ingreso de Reclamo
- Cancelar Reclamo
- Seguimiento de Reclamo
- Registro de Horas
- Liquidación de Haberes

Se pide:

- Realizar los diagramas de comunicación exclusivamente para las operaciones del sistema de los casos de uso antes mencionados.
- Mostrar en los diagramas el tipo de visibilidad que existe sobre el objeto al que se le envía un mensaje.