

Programación Avanzada

PRÁCTICO 4

Parte 1: Generación Parcial de Código

Ejercicio 1 (básico, imprescindible)

- a) Generar un esqueleto del código de las clases a partir de la estructura del Modelo de Dominio del ejercicio 4 del Práctico 3 - parte 1.
- b) Generar un esqueleto del método `main()` a partir del comportamiento contenido en el Diagrama de Secuencia del Sistema de la solución de la parte a) de dicho ejercicio.

Ejercicio 2 (básico, imprescindible)

- a) Generar un esqueleto del código de las clases a partir de la estructura del Modelo de Dominio del ejercicio 5 del Práctico 3 - parte 1.
- b) Generar un esqueleto del método `main()` a partir del comportamiento contenido en el Diagrama de Secuencia del Sistema de la solución de la parte a) de dicho ejercicio.
- c) Discutir según diferentes alternativas de Diagramas de Secuencia del Sistema para el caso de uso descrito en la parte a) de dicho ejercicio.

Ejercicio 3 (avanzado, imprescindible)

- a) Generar un esqueleto del código de las clases a partir de la estructura del Modelo de Dominio de la solución del ejercicio 14 del Práctico 2 - parte 1.
- b) Generar un esqueleto del método `main()` a partir del comportamiento contenido en los Diagramas de Secuencia del Sistema de la solución del ejercicio 8 del Práctico 3 - parte 1.

Parte 2: Conceptos Básicos

Ejercicio 4 (básico, imprescindible)

Considerar definidas las clases A, B, C, D y E, que cumplen $B <: A$, $C <: A$, $D <: B$ y $E <: D$.

Suponer que A tiene una operación polimórfica $g()$ que se encuentra redefinida en el resto de las clases menos en D y tiene otra operación $h()$ no polimórfica. Discutir qué método se ejecuta dependiendo de la instancia, tipo de despacho y operación invocada.

Ejercicio 5 (básico, imprescindible)

Considerar definidas las clases D, E y F, que cumplen $F <: E$, y una operación global con la siguiente signatura.

```
void g(E param)
```

Considerar además que se cuenta con los objetos d, e y f, instancias directas de las clases D, E y F respectivamente.

Indicar cuáles de las siguientes invocaciones son válidas. Justificar.

- a) $g(d)$
- b) $g(e)$
- c) $g(f)$

Ejercicio 6 (avanzado, de práctica)

- a) ¿Los términos método y operación denotan el mismo concepto? Justificar.
- b) Identificar clases, atributos, operaciones y métodos en la siguiente descripción del diseño de una parte de un sistema de gestión de personal.

Se considera una empresa que mantiene información de sus empleados. De cada empleado se almacena su nombre y se cuenta con una operación para obtener el mismo. Los empleados de la empresa pueden ser comunes, de los cuales se conoce su sueldo, o jornaleros, de los cuales se conoce la cantidad de horas que trabajó y el valor de la hora. Todos los meses, al realizarse la liquidación de los sueldos, la empresa calcula el monto total por concepto de sueldos, en base al sueldo fijo para empleados comunes y como la cantidad de horas por el valor de la hora para empleados jornaleros.

- c) ¿Existe alguna clase abstracta en su solución? ¿En qué situaciones debemos definir a una clase como abstracta?
- d) Dar una descripción (pseudocódigo) de la operación `getTotal()` mediante la cual la empresa calcula el monto total de la liquidación de todos sus empleados. Tener en cuenta que en un futuro pueden incorporarse nuevos tipos de empleados, y que la repercusión en la implementación existente debe ser mínima (en particular sobre las operaciones ya implementadas en clases existentes).
- e) Definir el concepto de polimorfismo e identificar su aplicación en el modelo realizado en la parte (b). ¿Qué beneficios le aportó el polimorfismo y cuál otra propiedad le fue de utilidad?
- f) Implementar en C++ el modelo anterior incluyendo la funcionalidad de obtener el total de sueldos.

- g) Realizar un procedimiento `main()` que permita, mediante un menú, ingresar una lista de empleados y que calcule el total de sueldos a pagar a los empleados de la empresa.

Ejercicio 7 (medio, imprescindible)

- a) Definir y explicar la utilidad de los conceptos interfaz y realización.
- b) Identificar clases, interfaces y realizaciones en las siguientes realidades.
- i) Un estudio de mercado realizado por una marca de lavarropas determinó la construcción de un modelo del mencionado electrodoméstico, el cual debe tener un conjunto de controles determinado C. La compañía fabricante decidió construir una cantidad de lavarropas, de los cuales una cierta cantidad son fabricados con componentes de procedencia P1 y el resto con componentes de procedencia P2. A la vista de los usuarios finales, todos los lavarropas son iguales.
 - ii) Un reproductor DVD's utiliza como salida de audio la de una televisión, pero en ocasiones cuando se quiere mejorar el sonido éste se conecta directamente a la salida de un equipo de audio, para lo cual se requiere que tanto la conexión con la televisión como con el equipo de audio sean la misma, si bien difieren en el mecanismo de la reproducción del sonido.
 - iii) Un equipo de audio cuenta con salidas de sonido, a las cuales se accede a través de conexiones que utilizan el tipo de ficha f1. De esta forma se pueden conectar parlantes de tipo P1 o P2 utilizando el tipo de ficha mencionado. Un sistema de sonido está compuesto por el equipo de audio y sus parlantes.
 - iv) Durante el desarrollo de un determinado software, se ha detectado la necesidad de contar con un módulo de manejo de colecciones de elementos. Las operaciones necesarias son: crear, agregar, eliminar, chequeo de vacío, buscar un elemento y operaciones de iteración sobre la colección. Se cuenta con tres implementaciones eficientes de estructuras de datos para colección: lista enlazada, árbol binario y tabla de dispersión. El módulo pretende poder manejar todos los tipos de colecciones sin importar realmente su implementación.
- c) ¿Qué pasaría si no contáramos con los conceptos interfaz y realización en las cuatro realidades planteadas?

Parte 3: Diagramas de Comunicación

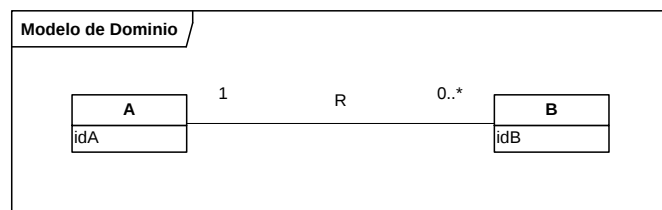
Ejercicio 1 (básico, imprescindible)

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*, el cual 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 (básico, imprescindible)

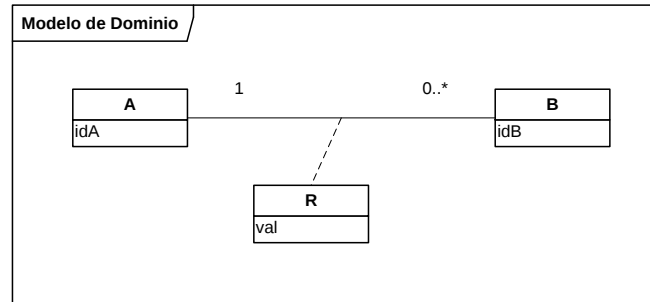
Considerar el Modelo de Dominio presentado en el siguiente diagrama.



- 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.

Ejercicio 3 (básico, imprescindible)

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.

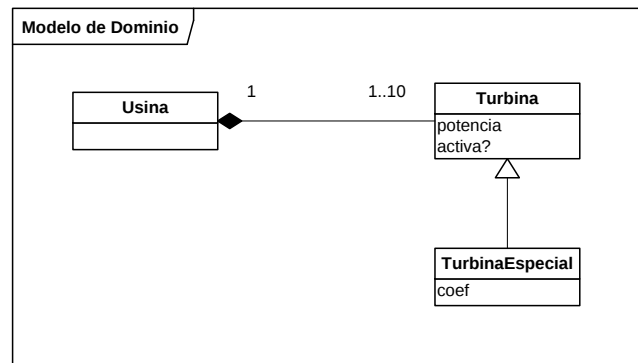
- Realizar el diagrama de comunicación sabiendo que *R* es navegable de *A* hacia *B*.
- Realizar el diagrama de comunicación sabiendo que *R* es navegable de *B* hacia *A*.
- Realizar el diagrama de comunicación asignando la responsabilidad a *R*.

Ejercicio 4 (básico, imprescindible)

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 (básico, imprescindible)

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 (medio, imprescindible)

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
----------------------	-----------------------

...

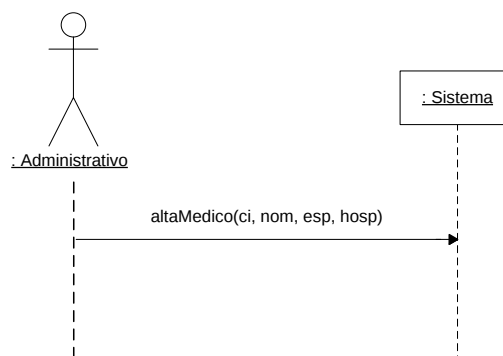
Nombre Empresa B

...

- 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 (básico, imprescindible)

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.

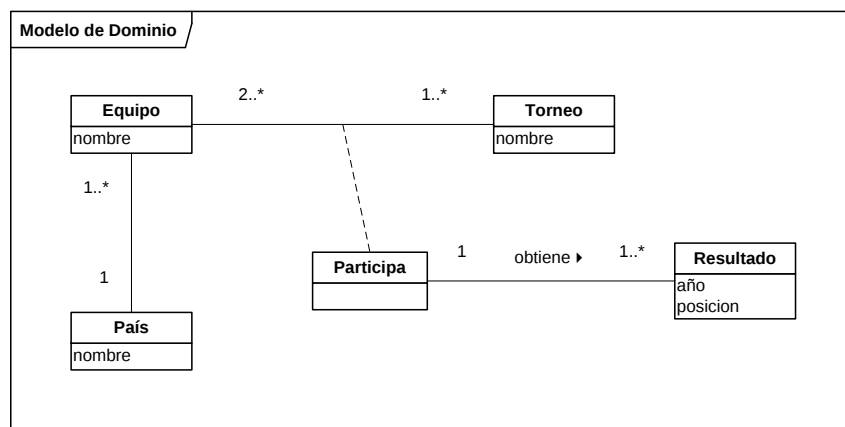


Nombre	Alta Médico
Operación	<code>altaMedico(ci:String, nom:String, esp:String, hosp:String)</code>
Entrada	<i>ci</i> la cédula del médico, <i>nom</i> el nombre del médico, <i>esp</i> la especialidad del médico, <i>Hosp.</i> identificador del hospital en el que trabaja el médico.
Descripción	Dar de 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> esta 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 <i>hosp</i> .	

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

Ejercicio 8 (medio, imprescindible)

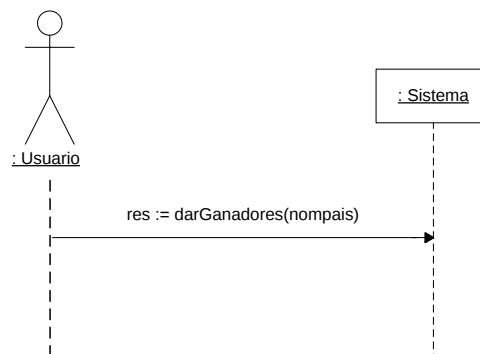
Considerar el siguiente Modelo de Dominio.



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."

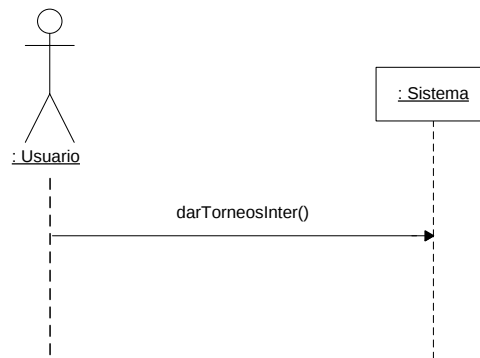
Para este caso de uso, se realizaron los siguientes artefactos durante el análisis.



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.	
post: El conjunto resultado equivale a los nombres de todos los equipos del país dado 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.

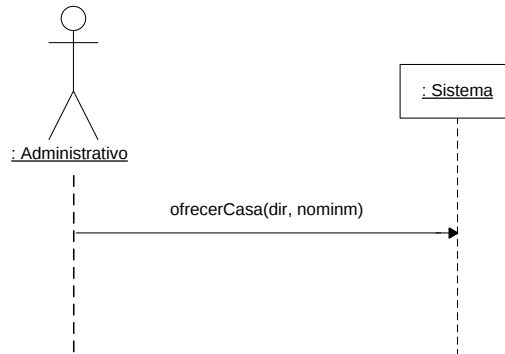


Nombre	Dar Torneos Internacionales
Operación	<code>darTorneosInter():Set(String)</code>
Salida	El conjunto de nombres de torneos internacionales.
Descripción	Devuelve el conjunto de nombres de torneos internacionales.
Pre- y poscondiciones	
post: El conjunto resultado equivale a los nombres de todos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

Ejercicio 9 (medio, de práctico)

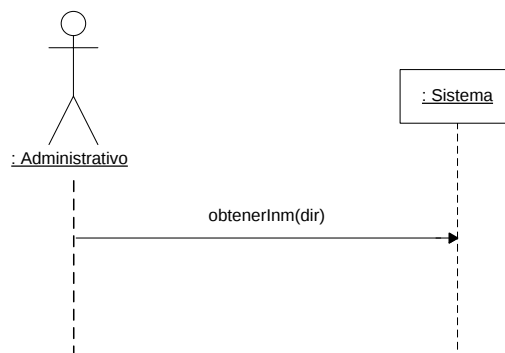
Considerar la realidad del Ejercicio 9 del Práctico 2 referente a negocios inmobiliarios y los artefactos de análisis correspondientes a los casos de uso Ofrecer Casa, Obtener Inmobiliarias y Agendar Visita. Diseñar las interacciones para las colaboraciones que realizan estos casos de uso.

a) Caso de uso Ofrecer Casa



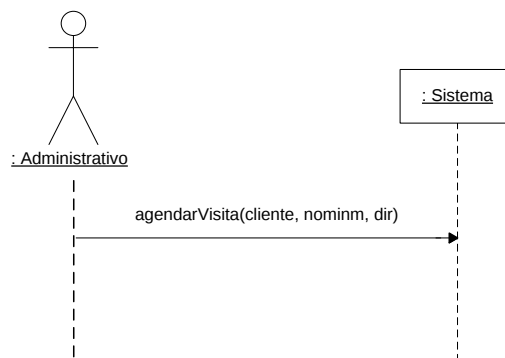
Nombre	Ofrecer Casa
Operación	<code>ofrecerCasa(dir:String, nominm:String)</code>
Entrada	<i>dir</i> es la dirección de la casa que se desea ofrecer, <i>nominm</i> es el nombre de la inmobiliaria que ofrecerá la casa.
Descripción	Registrar que la casa ubicada en <i>dir</i> sea ofrecida por la inmobiliaria <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 y la inmobiliaria.	

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



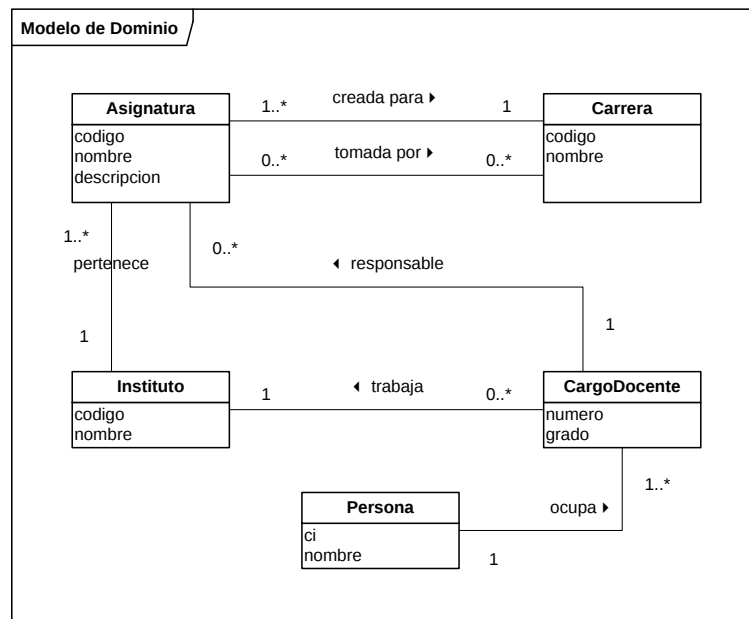
Nombre	Agendar Visita
Operación	<code>agendarVisita(cliente:String, nominm:String, dir:String)</code>
Entrada	<i>cliente</i> es el nombre del cliente, <i>nominm</i> es el nombre de la inmobiliaria y <i>dir</i> es la dirección de la casa.
Descripción	Agendar la visita del cliente a la casa a través de 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 existe en el sistema.	
post: Se registró que el cliente visitará la casa a través de la inmobiliaria.	

Ejercicio 10 (avanzado, imprescindible)

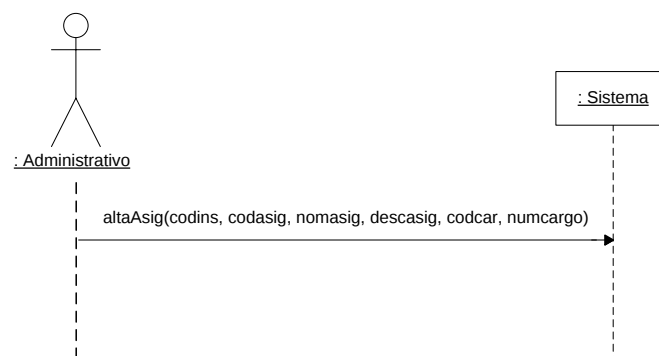
A partir del siguiente documento de Visión del Problema se construyó el Modelo de Dominio correspondiente.

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, numcarga:Integer)</code>
Entrada	<i>codins</i> es el código del instituto, <i>codasig</i> es el código de la asignatura, <i>nomasig</i> es el nombre de la asignatura, <i>descasig</i> es la descripción de la asignatura, <i>codcar</i> es el código de la carrera y <i>numcarga</i> es el número del cargo docente responsable.
Descripción	Da de alta una asignatura en el sistema
Pre- y poscondiciones	
<pre> pre1: El instituto codins existe en el sistema. pre2: El cargo docente numcarga existe en el sistema. pre3: La carrera codcar existe en el sistema. pre4: No existe en el sistema una carrera con código codcar. post1: Existe una nueva instancia de Asignatura en el sistema cuyos atributos llevan los valores codasig, nomasig y descasig. post2: La nueva asignatura y el instituto codins están linkeados. post3: La nueva asignatura y la carrera codcar están linkeados por la asociación creada para. post4: La nueva asignatura y el cargo docente numcarga 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 (avanzado, de práctica) :: Laboratorio 2008

Considerando la realidad presentada en prácticos anteriores, en esta etapa se debe diseñar las operaciones de 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.

Parte 4: Laboratorio

Observaciones:

- La entrega podrá realizarse hasta el **domingo 03 de junio a las 23:59 hs.**
- Para realizar la entrega se deberá completar el informe de casos de uso disponible en la web.
- Las entregas serán vía mail y deberán consistir en un único archivo comprimido .zip con el formato lab04grXXB.zip (buceo) / lab04grXXL.zip (latu) teniendo que ser enviados a las direcciones progavan.buceo@gmail.com y progavan.latu@gmail.com respectivamente.
- El asunto del mail deberá ser "Entrega laboratorio 04 Grupo XX - v1.0", las sucesivas entregas que se realicen sobre este laboratorio deberán modificar la versión (v2.0, v3.0, etc.)
- Las entregas que no cumplan estos requerimientos no serán consideradas. El hecho de no realizar una entrega implica la insuficiencia del laboratorio completo.

Considerando la realidad presentada en los ejercicios de laboratorio anteriores, en esta etapa se debe diseñar las operaciones de sistema detectadas en la etapa anterior para los siguientes casos de uso:

- Crear Mapa
- Eliminar Mapa
- Crear Partida
- Jugar Turno de Partida

Se pide

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