

# Programación 4

## EXAMEN DICIEMBRE 2011

Por favor siga las siguientes indicaciones:

- Escriba con lápiz.
- Escriba las hojas de un solo lado.
- Escriba su nombre y número de documento en todas las hojas que entregue.
- Numere las hojas e indique el total de hojas en la primera de ellas.
- Recuerde entregar su número de examen junto al examen.

### Problema 1 (30 puntos)

Una nueva empresa ha recibido financiamiento por parte de un inversor para llevar adelante su idea en un prototipo. La misma consiste en una aplicación para la plataforma iOS (es decir para iPhone, iPad y iPod Touch) que permite al usuario administrar calendarios. Si bien esta plataforma ya cuenta con una aplicación nativa para manejo de agenda, la nueva aplicación a construir tendrá algunos diferenciales.

El siguiente folleto describe las funcionalidades principales:

***iEvents***

¿Quisiera manejar múltiples calendarios pero la aplicación que utiliza no se lo permite? ¿Necesita diferenciar los contactos personales de los empresariales? ¿Desea que las reuniones envíen mail automáticamente a los invitados? ¿Quiere todo esto y mucho más en forma gratuita?

iLe presentamos ***iEvents App***! La nueva aplicación para iOS que revolucionará la forma en que maneja su calendario... para siempre!

**¿Qué es?**  
***iEvents App*** es una aplicación que le permite llevar múltiples calendarios con sus respectivos eventos y contactos.

**¿Qué características tiene?**  
Permite el manejo de contactos, recordatorios, tareas y reuniones de forma integrada en una misma pantalla. Diferencie los contactos personales de los contactos empresariales, agende reuniones con invitados (y ***iEvents App*** les enviará mail en forma automática), agende tareas para hacer (incluyendo una descripción de la misma), agende recordatorios con o sin alarma y asociados o no a tareas o reuniones.

**¿Cuáles son las funcionalidades?**  
***iEvents App*** le permite:

- \* Crear calendario (con un nombre que lo identifica y contactos)
- \* Agregar contactos a un calendario (empresariales o personales, con el mail que los identifica, un nombre y teléfonos)
- \* Agendar tareas a un calendario (con su descripción, título, fecha y hora)
- \* Agendar reuniones a un calendario (con su motivo, título, fecha y hora, lugar, lista de invitados elegidos entre los contactos del calendario y si es una reunión periódica indicar la periodicidad: semanal, mensual, anual)
- \* Agendar recordatorios de tareas y reuniones a un calendario (con su título, fecha y hora, si lleva alarma o no y si es un recordatorio de una tarea ó de una reunión)

\* Visualizar los calendarios de forma unificada: en una misma ventana muestra todos los eventos (tareas y reuniones) de todos los calendarios definidos por el usuario.

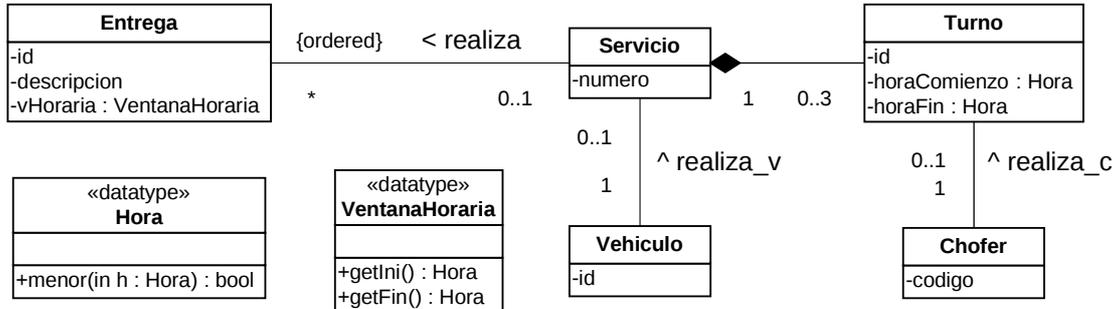
**Se pide:**

- i. Realice el Modelo de Dominio de la realidad anterior, incluyendo restricciones en lenguaje natural.
- ii. A partir del siguiente Caso de Uso realice el DSS correspondiente incluyendo datatypes en caso de ser necesarios.

<b>Caso de Uso</b>	Agendar Reunión
<b>Actor</b>	Usuario
<b>Descripción</b>	Este caso de uso comienza cuando el usuario visualiza todos los calendarios y selecciona el calendario en el cual desea agregar la nueva reunión. Luego indica el título, fecha, hora y ubicación (lugar donde se realizará la reunión) y elige uno por uno, de la lista de contactos, los contactos que serán invitados a la reunión (y que la aplicación enviará mail automáticamente). En el caso en que un invitado ya tenga agendado algún evento para ese día/hora, el sistema informará de esta situación y preguntará al usuario si de todos modos desea invitar a ese contacto o no. Finalmente, el usuario indica la periodicidad de la reunión, si es que la tiene, entre semanal, mensual o anual (se asume que se reitera el día y la hora).

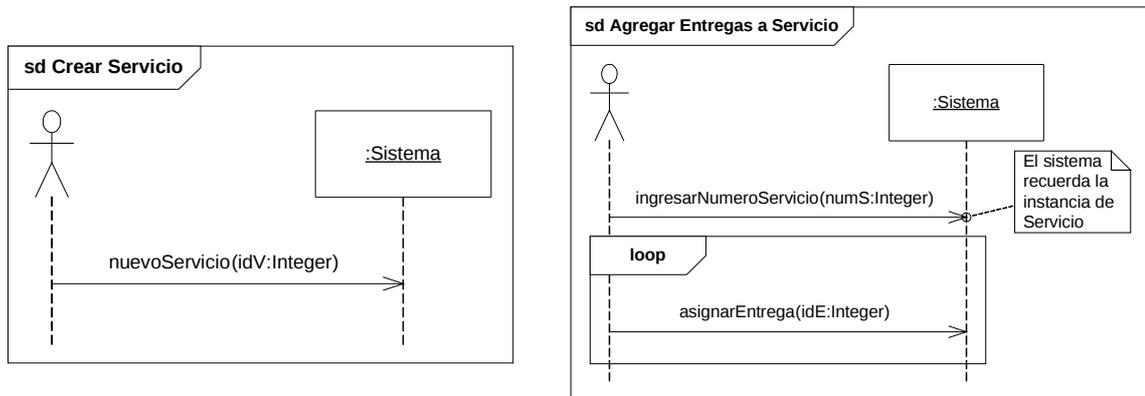
## Problema 2 (35 puntos)

La sección distribución de una compañía de servicios, se encuentra en proceso de elaboración de un software de apoyo a la planificación de su operativa diaria. Dicha operativa consiste en realizar un conjunto de entregas, que son asignadas a un servicio (ruta que realiza un vehículo durante todo un día), que a su vez está dividido en varios turnos (cada uno asociado a un chofer que maneja el vehículo correspondiente al servicio). La figura muestra un modelo conceptual realizado para representar la realidad descrita.



### PARTE A

Se cuenta con diagramas de secuencia del sistema de los casos de uso **Crear Servicio** y **Agregar Entregas a Servicio**, junto con los contratos parciales de las operaciones del sistema asociadas a dichos casos.



<b>Operación</b>	nuevoServicio(idV:Integer);
<b>Pre y postcondiciones</b>	
<b>pre:</b>	Existe en el sistema un Vehiculo, cuyo atributo id tiene el valor idV.
<b>post:</b>	Existe una nueva instancia de Servicio, cuyo atributo num coincide con el último número de servicio disponible en el Sistema.
<b>post:</b>	Se incrementa en uno el valor del último número de servicio disponible en el Sistema.
<b>post:</b>	Existe un nuevo link entre la instancia creada de Servicio y la instancia de Vehiculo identificada por idV.

<b>Operación</b>	ingresarNumeroServicio(numS:Integer);
<b>Pre y postcondiciones</b>	
<b>pre:</b>	Existe en el sistema un Servicio, cuyo atributo numero tiene el valor numS.
<b>post:</b>	Existe en la memoria del sistema un Servicio, cuyo atributo numero tiene el valor numS.

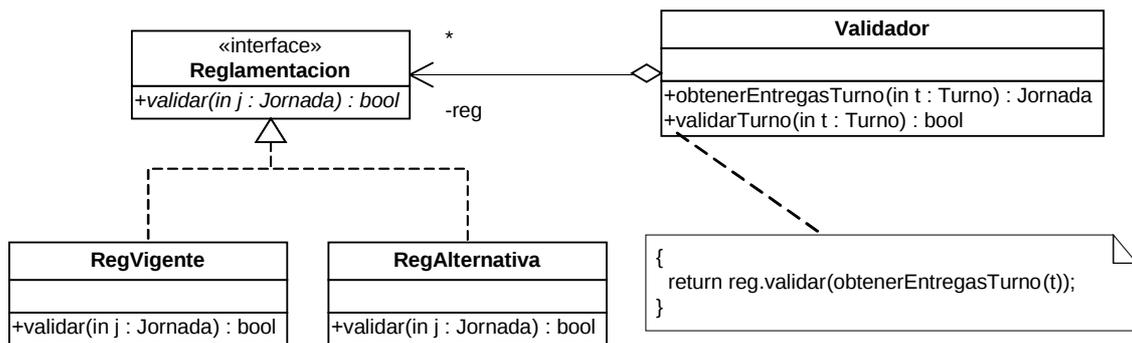
<b>Operación</b>	asignarEntrega(idE:Integer);
<b>Pre y postcondiciones</b>	
<b>pre:</b> Existe en el sistema una Entrega, cuyo atributo id tiene el valor idE.	
<b>pre:</b> Existe en la memoria del sistema un Servicio (denominado serv).	
<b>post:</b> Existe un nuevo link entre la Entrega cuyo atributo id tiene el valor idE y el Servicio serv.	

**Se pide:**

- i. Realizar los Diagramas de Comunicación de las 3 operaciones del sistema.
- ii. Realizar el Diagrama de Clases de Diseño resultante.

**PARTE B**

Los Turnos realizados por los Choferes deben respetar ciertas reglamentaciones laborales, como ser disponibilidad de tiempo para descanso entre Entregas, holgura en las ventanas de tiempo de las diferentes Entregas, etc. El sistema maneja diferentes reglamentaciones (por ejemplo la vigente, la anterior o una reglamentación alternativa que se piensa implementar). Las reglamentaciones se organizan según el diagrama de la figura.



**Se pide:**

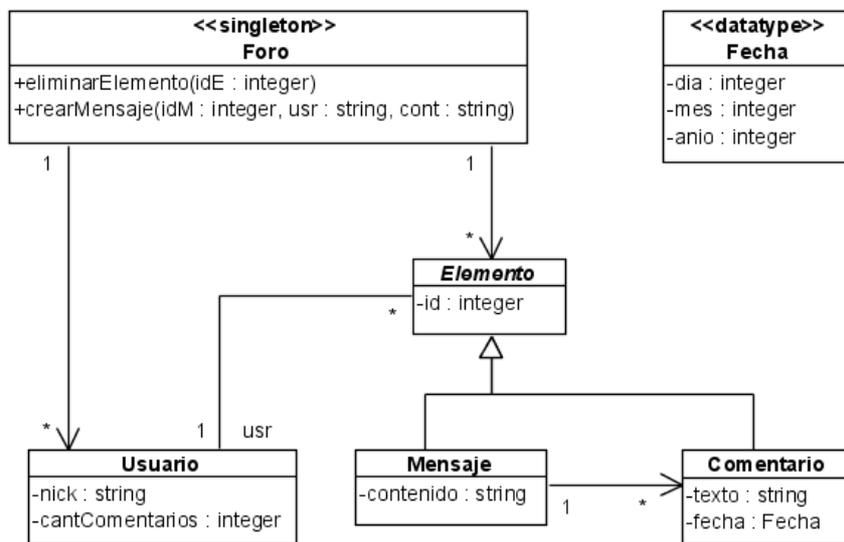
- i. Realizar el Diagrama de Comunicación de la operación **obtenerEntregasTurno**, que devuelve una instancia de clase Jornada (cuyos detalles no se especifican), conteniendo las Entregas del Servicio asociado al Turno recibido como parámetro, cuya ventana horaria está contenida entre las horas de comienzo y fin de dicho turno. Asumir que cuenta con una operación denominada **armarJornada**, que recibe un Set(Entrega) y devuelve una Jornada formada por dichas Entregas.
- ii. Nombrar el patrón de diseño utilizado para la validación de turnos, indicando los roles de cada clase.

### Problema 3 (35 puntos)

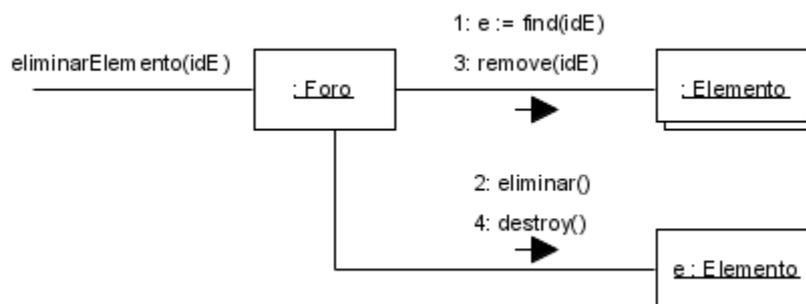
Se le ha encomendado la implementación de un foro de mensajes que se utilizará como un componente durante la construcción de un sitio web donde los usuarios pueden compartir información interesante como links a páginas web u otros contenidos de interés. Para esto el sistema mantiene una lista de usuarios activos que pueden agregar nuevos mensajes en el foro y comentar sobre ellos. De cada mensaje se conoce su contenido y los comentarios de los usuarios. De cada usuario se conoce su nick y la cantidad de comentarios que realizó. De cada comentario se conoce su texto y la fecha en la cual fue realizado.

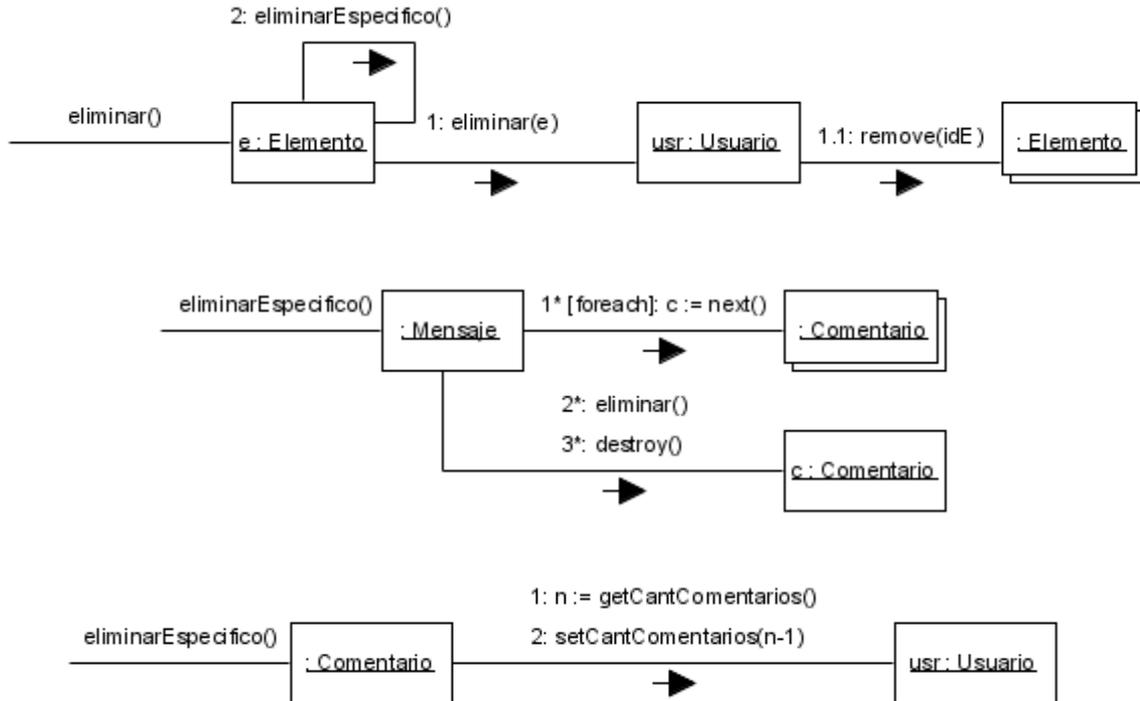
Por una decisión tomada en la etapa de diseño, tanto los mensajes como los comentarios se identifican por un id, es decir, no pueden existir dos mensajes o comentarios con el mismo id. Para cada usuario se almacena los comentarios y mensajes que ingresó en el foro.

El equipo de desarrollo generó el siguiente diagrama de clases **parcial** que se deberá seguir en la implementación.



En primer lugar es necesario dar una implementación para la operación **eliminarElemento()** que permite quitar un mensaje o un comentario del foro. Si es un mensaje, se eliminan todos los comentarios asociados, y si es un comentario, se actualiza el valor de la variable cantComentarios del usuario que generó ese comentario. A continuación se brinda un diagrama de comunicación para la operación.





También se necesita implementar la operación **crearMensaje(idM, usr, cont)** que crea una instancia de mensaje con *idM* como valor del atributo id y *cont* como su contenido. Además, esta operación agrega esa instancia al conjunto de elementos del usuario con nick *usr* y también al conjunto de elementos del foro.

Dado que el código que usted genere se va a reutilizar en otros componentes del sistema, el equipo de diseño requiere que la clase **Elemento** **no contenga** operaciones de tipo “setters”..

**Se pide**

- i. Escribir el código completo (.h y .cpp) de las clases **Foro**, **Elemento**, **Mensaje** y **Comentario**, conteniendo la implementación de las operaciones descritas.
- ii. Escribir la definición del Datatype **Fecha** (solo .h) incluyendo los siguientes operadores donde *f1* y *f2* son objetos de tipo Fecha.
  - Comparación entre fechas > (*f1* > *f2*)
  - Asignación de fechas (*f1* = *f2*)
  - Inserción de flujo (cout << *f1*)

**Notas**

- No incluir la definición de operaciones de acceso (**get** y **set**)
- No incluir código de preprocesador
- Suponer que existe el datatype **String** que almacena una cadena de caracteres
- Suponer que existe las clases **Col** y **Dict** que implementan las interfaces **ICollection** e **IDictionary** respectivamente, con implementaciones de **Iterator** para cada caso.
- Suponer que existen las clases **KeyString** y **KeyInteger** que implementan **IKey** y permiten almacenar claves de tipo **String** y entero respectivamente.