

Programación 4

PARCIAL FINAL EDICIÓN 2018

Cantidad total de páginas: 5

Por favor siga las siguientes indicaciones:

- Escriba con lápiz y de un solo lado de las hojas
- 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 parcial junto al parcial
- Está prohibido el uso de computadoras, tabletas o teléfonos durante el parcial

Problema 1 (35 puntos)

Parte I)

¿Sobre qué términos se expresan las PRE y POST condiciones de una operación en un Contrato de Software?

Parte II)

Una incubadora de empresas es una organización diseñada para acelerar el crecimiento y asegurar el éxito de proyectos emprendedores a través de una amplia gama de recursos y servicios empresariales como: renta de espacios físicos, capitalización, consultorías, networking, telecomunicaciones, limpieza, etc. [Wikipedia] En Uruguay existen varias incubadoras (e.g. Ingenio, Sinergia, Khem, Da Vinci) las cuales generalmente proveen apoyo de consultores a los emprendimientos incubados, siendo los consultores remunerados por las horas dedicadas.

Se desea desarrollar un Sistema que lleve el registro de este apoyo, de forma de facilitar la liquidación de horas a los consultores.

Todos los emprendimientos tienen un nombre (que lo identifica) así como una fecha de creación y consta de un conjunto de emprendedores (de los cuales interesa su nombre, cédula y fecha de nacimiento). Asimismo, los emprendimientos pueden ser de Productos (ej: videojuegos) o de Servicios (ej: desarrollo a medida), no pudiendo ser de ambos.

Por otro lado, los consultores se identifican por su número de RUT, se especializan ya sea en Productos o Servicios (sólo una de esas especializaciones), y son asignados a emprendimientos (los cuales apoyarán con su experiencia), siempre que sean emprendimientos del tipo que se especializan.

El valor-hora de un consultor se establece por paquete de reuniones. Un paquete se lo identifica por un nombre. Si bien un paquete es un conjunto de reuniones, el paquete no tiene prefijado el conjunto de reuniones sino que las reuniones se van definiendo en el transcurso del apoyo que brinda el consultor. Por lo tanto, un paquete tendrá al menos una reunión.

Se debe mantener el registro de reuniones de cada consultor en sus paquetes. De la reunión interesa su fecha, la duración (en horas) y los emprendedores que participaron de la reunión con el consultor. De esta forma se podrá luego recorrer el registro de cada consultor para saber las reuniones que mantuvo y poder sumarlas para obtener el total a pagar al consultor.

Se cuenta además con el siguiente Caso de Uso de Alto Nivel:

Caso de Uso:	Alta de Reunión
--------------	-----------------

Actor:	Usuario
Sinopsis:	Este Caso de Uso comienza cuando el Usuario elige crear una nueva reunión, ingresando la fecha, la duración (en horas) de la misma y el nombre del paquete a la que corresponde. Luego, el Sistema le muestra una lista de todos los emprendimientos, de los cuales el Usuario debe elegir uno. Para ese emprendimiento elegido, el Sistema le devuelve la lista de todos los emprendedores y el Usuario deberá seleccionar todos aquellos que participaron de la reunión. A continuación, el Sistema muestra una lista de todos los consultores asociados al emprendimiento, debiendo el Usuario elegir el consultor que apoyó a esos emprendedores en esa reunión. Finalmente, el Usuario confirma el alta de la nueva reunión al Sistema. El Sistema crea la nueva reunión y la agrega al paquete si es que existe, de lo contrario crea también un nuevo paquete con valor-hora en 0.

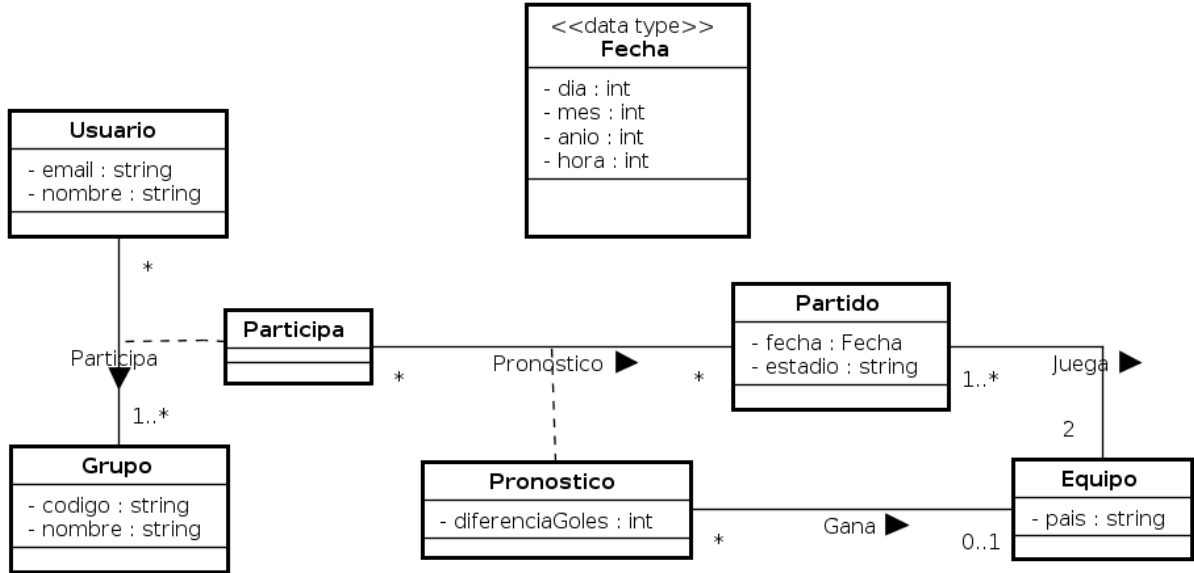
Se pide:

- Realizar el Modelo de Dominio de la realidad planteada en la descripción y el caso de uso, con restricciones en lenguaje natural.
- Realizar el DSS para el Caso de Uso Alta de Reunión, indicando el uso de Memoria del Sistema así como de Datatypes.

Problema 2 (35 puntos)**Parte I)**

Una empresa ha lanzado una aplicación llamada *pencaFing*, que permite realizar predicciones sobre el resultado de los partidos del mundial de Rusia. Los usuarios pueden registrarse en la aplicación, y participar en varios grupos. En cada uno de estos grupos podrán ingresar un pronóstico por partido, en donde indicarán el equipo ganador y la diferencia de goles del partido.

Los usuarios serán identificados dentro de la aplicación por su email, y los grupos por un código alfanumérico que podrá ser utilizado por los usuarios para invitar a otros a grupos.



Se pide:

Realizar el Diagrama de Comunicación, **incluyendo visibilidades** de las siguientes dos (2) operaciones. Para el caso de *obtenerPronóstico*, definir el data type DTProntico.

<pre>crearUsuario(String email, String nombreUsuario, String codigoGrupo)</pre>	
Descripción	Crea al usuario y lo agrega al grupo con código <i>codigoGrupo</i>
Parámetros	<ul style="list-style-type: none"> ❖ email: Email del usuario a crear ❖ nombreUsuario: Nombre del usuario a crear ❖ codigoGrupo: Código que identifica al grupo
Precondiciones	<ul style="list-style-type: none"> ❖ No existe en el sistema una instancia de Usuario con email <i>email</i>. ❖ Existe en el Sistema una instancia de Grupo con código <i>codigoGrupo</i>.
Postcondiciones	<ul style="list-style-type: none"> ❖ Se crea una instancia de Usuario con email <i>email</i> y nombre <i>nombreUsuario</i>. ❖ Se crea una instancia del tipo Asociativo Participa linkeado a la instancia de Usuario creada y a la instancia de Grupo cuyo código es <i>codigoGrupo</i>

<pre>obtenerPronosticos(String email, String codigoGrupo): Set(DtPronostico)</pre>	
Descripción	Se obtiene información de los pronósticos de un usuario en un grupo
Parámetros	<ul style="list-style-type: none"> ❖ email: mail del usuario ❖ codigoGrupo: Código que identifica al grupo
Precondiciones	<ul style="list-style-type: none"> ❖ Existe en el sistema una instancia de Usuario con email <i>email</i>. ❖ Existe en el Sistema una instancia de Grupo con código <i>codigoGrupo</i>.
Postcondiciones	<ul style="list-style-type: none"> ❖ Se retorna y construye un conjunto de data types <i>DtPronostico</i>, con la información de las instancias de <i>Pronóstico</i> que tienen un link con el tipo asociativo <i>Participa</i>

	<p>entre la instancia de Usuario con email <i>email</i> y la instancia de Grupo con código <i>codigoGrupo</i>. La información que contendrá cada DtPronostico es:</p> <ul style="list-style-type: none"> ➤ Fecha del partido ➤ Diferencia de goles ➤ Países de los equipos que juegan el partido ➤ Si no es empate, el país del equipo ganador
--	--

Parte II)

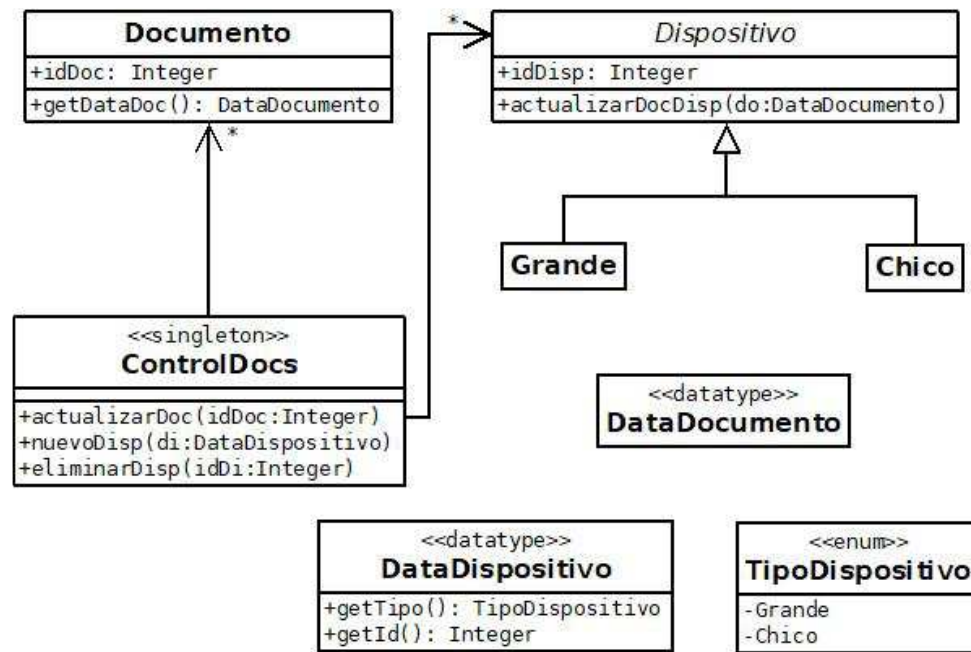
Se quiere que cada uno de los participantes de un grupo sea notificado cada vez que un nuevo usuario se une a dicho grupo. A cada usuario se le deberá enviar el email del nuevo participante y el código del grupo al que se unió.

Se pide:

Realizar el Diagrama de Clases de Diseño (DCD) para soportar todo lo requerido en las partes I y II. En caso de utilizar algún patrón de diseño indicar su nombre, y qué rol cumple en el patrón cada clase involucrada.

Problema 3 (30 puntos)

La figura muestra el Diagrama de Clases de Diseño (DCD) parcial del manejador central de un sistema de almacenamiento de documentos en la nube, que ofrece la posibilidad de mantener réplicas locales de los documentos en diferentes dispositivos. Esto permite por ejemplo, crear un documento utilizando un software instalado localmente en un dispositivo A (por ejemplo, el laptop), almacenarlo en la nube y luego acceder al documento desde otro dispositivo B (por ejemplo, tablet o teléfono inteligente), pudiendo trabajar con el documento de forma local. Cada dispositivo tiene una copia local de los documentos y cada vez que un documento se modifica en alguno de los dispositivos, todos los otros dispositivos deben actualizar su respectiva copia local. Los dispositivos pueden ser de diferente tipo, según su capacidad de procesamiento local.



Las operaciones del controlador `ControlDocs` realizan las siguientes acciones:

- `actualizarDoc`: Obtiene el documento a partir de su identificador y obtiene un datavalue con sus datos, a partir del cual actualiza el documento en cada uno de los dispositivos.
- `nuevoDisp`: Crea el dispositivo a partir del parámetro y lo agrega a la colección de dispositivos.
- `eliminarDisp`: Elimina el dispositivo del sistema.

Se pide:

- Implementar en C++ completamente los .h de todas las clases del DCD. No implementar los datatypes ni el enumerado.
- Implementar en C++ completamente el .cpp de la clase `ControlDocs`.

Observaciones:

- Puede utilizar colecciones genéricas (realizaciones de `IDictionary` e `ICollection`) o paramétricas (contenedores STL).
- Implementar exclusivamente las operaciones del DCD. Incluir constructores y destructor cuando sea necesario.
- No incluir directivas al precompilador.