

Programación 4

PARCIAL FINAL EDICIÓN 2019

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 (30 puntos)

Un grupo de excompañeros de clase de la Fing deciden unirse para crear un pequeño emprendimiento de consultoría a otras empresas, aprovechando su experiencia y buena relación. En ese contexto, se le ha pedido a usted que modele un sistema para el registro de sus empresas cliente y las propuestas de trabajo que se les realizan.

De las empresas cliente interesa saber su nombre (que las identifica), dirección, cantidad de empleados y los contactos que trabajan en esa empresa (al menos uno por empresa, de quien se conocerá su nombre que lo identifica, su mail y su cargo).

Dado que no es lo mismo trabajar con una startup de Uruguay que con una empresa de USA, se ha definido la siguiente categoría de empresas: Startups (emprendimientos chicos en nuestro país), ProBono (empresas a las que no se les cobrará, esperando que en un futuro se conviertan en clientes rentables), LATAM (empresas de América Latina) y USA (empresas de USA). Cada categoría tiene su propio valor-hora en dólares, que es lo que cobrará el equipo consultor.

Finalmente, interesa registrar las propuestas de trabajo que se les realiza a cada empresa. Cada propuesta es para una empresa en particular, tiene un código que identifica a la propuesta, la fecha en la cual se realiza, un link al archivo PDF con el contenido de la misma y se debe indicar si la propuesta ya sido aprobada o no.

En cada propuesta pueden participar muchos consultores y se debe saber la cantidad de horas de cada consultor en esa propuesta (para luego poder pagarles según las horas que cada uno trabajó). De cada consultor interesa saber su nombre (que lo identifica), mail y su sueldo-hora.

Adicionalmente, se tiene la descripción del siguiente Caso de Uso.

Caso de Uso:	Alta de Nueva Propuesta
Actor:	Usuario
Sinopsis:	El caso de uso comienza cuando el usuario solicita un listado de todas las empresas cliente, para luego elegir aquella a la cual asociar la propuesta que se está ingresando. Luego el usuario solicita una lista de todos los consultores para poder elegir aquellos que participarán de la propuesta, incluyendo las horas que cada uno se prevé que trabaje. Luego de haber elegido el/los consultor/es, el usuario ingresa los datos de la propuesta (su código y el link al archivo PDF) siendo la fecha auto-calculada por el sistema; inicialmente la propuesta se almacena como no aprobada. Finalmente, el usuario decide confirmar o cancelar el ingreso de la nueva propuesta.

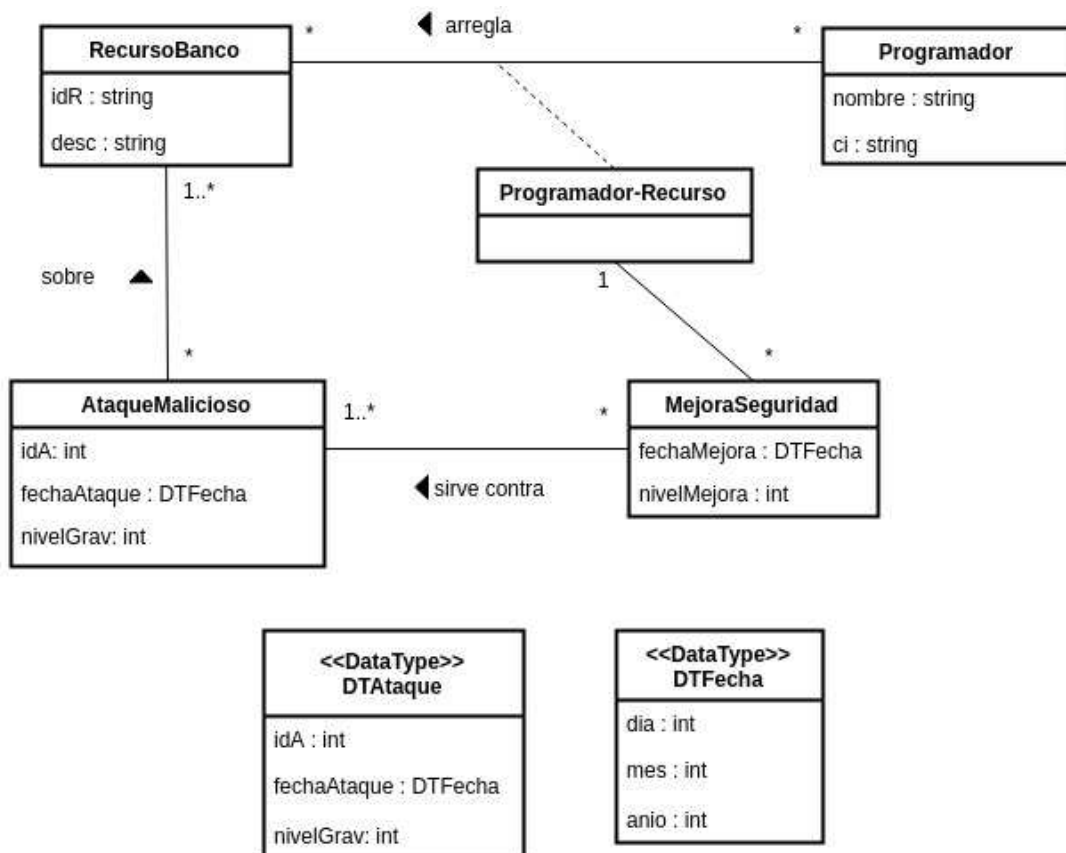
Se pide:

- Realizar el Modelo de Dominio de la realidad planteada, con restricciones en lenguaje natural.
- Realizar el Diagrama de Secuencia del Sistema para el Caso de Uso Alta de Nueva Propuesta, indicando el uso de Memoria del Sistema así como de Datatypes, si corresponde.

Problema 2 (35 puntos)

El banco *SantanFing* le ha encomendado a usted el mantenimiento de su sistema informático, en el que se tiene un registro de los recursos de extrema importancia (por ejemplo las cuentas bancarias y los datos de los clientes). El acceso indebido a estos recursos puede implicar la pérdida de mucho dinero, y el sistema es a menudo atacado por agentes maliciosos. Además de los recursos se registran esos ataques detectados, y se registran los programadores del banco que hacen mejoras de seguridad para que dichos ataques no sean exitosos.

Del recurso se almacena un identificador y una descripción. De los programadores se sabe el nombre, la cédula y las mejoras de seguridad que realizó sobre los recursos, con su fecha y su nivel. De las mejoras se sabe también sobre qué ataques son efectivas. Finalmente, de los ataques se mantiene un identificador, la fecha en la que se detectó y el nivel de daño que puede causar. La figura presenta el Modelo de Dominio construido en la etapa de análisis del sistema.



Se pide:

- a) Realizar los Diagramas de Comunicación correspondientes a las siguientes operaciones del sistema, **incluyendo visibilidades**.

obtenerAtaquesDadaUnaFecha(idR: string, f: DTFecha): Set(DTAtaque)	
Descripción	Retorna los ataques realizados sobre un recurso, registrados en la fecha indicada.
Parámetros	<ul style="list-style-type: none"> idR: El identificador en el sistema del recurso. f: Fecha en la que se quiere saber los ataques registrados sobre el recurso.
Precondiciones	<ul style="list-style-type: none"> Existe en el sistema un recurso con identificador idR.
Postcondiciones	<ul style="list-style-type: none"> Se retorna un set de elementos de tipo DTAtaque donde cada elemento representa un ataque sobre el recurso de identificador idR, y registrado en la fecha de valor f. Si no tiene ataques registrados en la fecha indicada, se retorna el set vacío.

Nota: Puede asumir que en la clase AtaqueMalicioso existe una función:
 mismaFecha(f: DTFecha): Bool que retorna True si el ataque se registró en esa fecha f, y False en caso contrario.

registrarMejora(idR: string, ciP: string, nvMejora: int, sirveContra: Set(int))	
Descripción	Registra una mejora de seguridad realizada por un programador, sobre un recurso.
Parámetros	<ul style="list-style-type: none"> idR: El identificador en el sistema del recurso. ciP: La cédula del programador registrado en el sistema. nvMejora: Nivel de la nueva mejora de seguridad. sirveContra: Set de identificadores de ataques maliciosos, contra los cuales la mejora sirve.
Precondiciones	Existen en el sistema: <ul style="list-style-type: none"> Un recurso con identificador idR. Un programador con cédula ciP. Una instancia de Programador-Recurso entre el recurso y el programador identificados por idR y ciP respectivamente. Para cada entero idA en el set sirveContra, existe un ataque registrado con identificador idA y asociado al recurso identificado por idR.
Postcondiciones	<ul style="list-style-type: none"> Se crea una nueva instancia de MejoraSeguridad con la fecha del sistema y nivel nvMejora. La nueva instancia se asocia a la instancia de Programador-Recurso entre el recurso y el programador identificados por idR y ciP respectivamente. Cada ataque de identificador perteneciente al conjunto sirveContra se asocia a la nueva instancia de mejora creada.

Nota: Puede asumir que el controlador que defina, tiene una operación
 obtenerFechaActual(): DTFecha, que retorna la fecha actual del sistema.

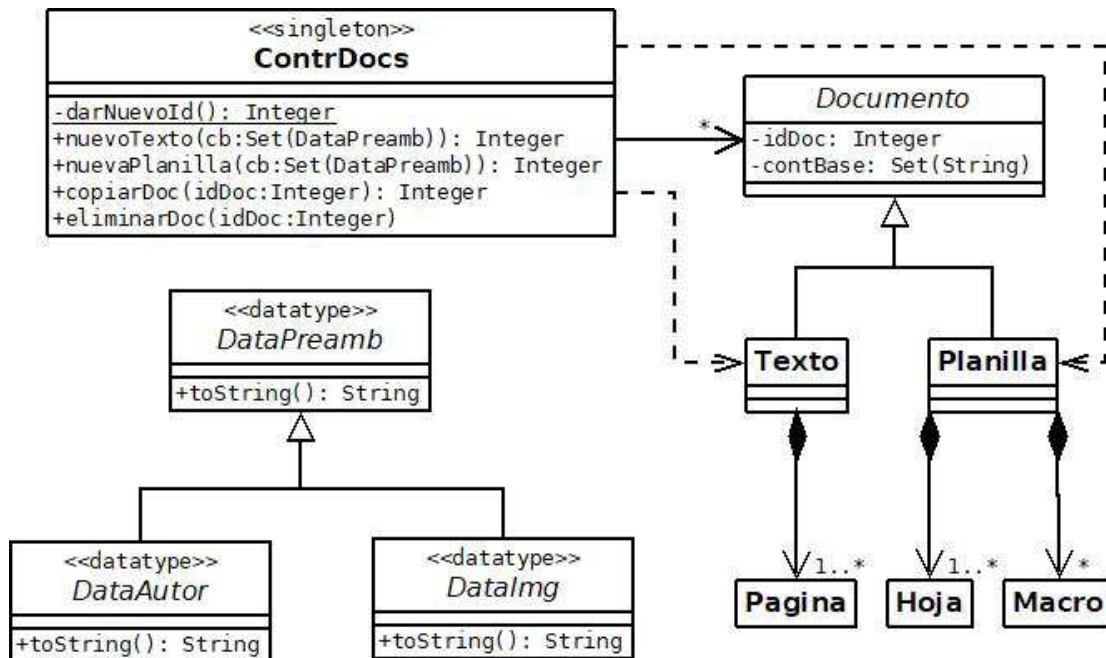
b) Realizar el Diagrama de Clases de Diseño (DCD) que soporte la parte anterior.

Nota: Incorporar en el DCD sólo los constructores, destructores, setters y getters que se utilicen en los Diagramas de Comunicación.

Problema 3 (30 puntos)

Se está construyendo un software de manejo de documentos que incluye archivos de texto y planillas de cálculo. La figura muestra un Diagrama de Clases de Diseño (DCD) parcial de la solución. Todos los documentos tienen un identificador y un contenido base (datos del autor, imagen de miniatura, etc) que es almacenado como un conjunto de cadenas de caracteres. Los archivos de texto están estructurados en varias páginas, mientras que las planillas de cálculo tienen varias hojas y además pueden tener macros (programas que ejecutan sobre los datos almacenados en las hojas). El controlador de documentos incluye las siguientes operaciones:

- `darNuevoId`: Retorna un identificador válido para un nuevo documento.
- `nuevoTexto`: Crea un nuevo archivo de texto en el sistema a partir de un contenido base y retorna su identificador. El contenido base es dado como una colección de datavalues de la jerarquía `DataPreamb`; la operación `toString` convierte el contenido del dato (cualquiera sea su representación interna) en una cadena de caracteres.
- `nuevaPlanilla`: Crea una nueva planilla de cálculo en el sistema a partir de un contenido base y retorna su identificador. Aplica la misma observación sobre el contenido base que para los documentos de texto.
- `copiarDoc`: Genera una copia en el sistema de un documento existente dado por su identificador y devuelve el identificador de la copia.
- `eliminarDoc`: Elimina del sistema un documento dado por su identificador.



Se pide: Implementar completamente (archivos .h y .cpp) las clases Documento, Texto, Planilla y ContrDocs, agregando todos los atributos y operaciones que no figuren en el DCD y que usted considere necesarias. Implementar los constructores, destructores, setters y getters que considere necesarios para resolver las operaciones pedidas. Tener en cuenta la aplicación de conceptos del curso para favorecer la reutilización de código y el ocultamiento de información.

Notas:

- Puede utilizar colecciones paramétricas (contenedores STL) o genéricas (realizaciones de ICollection e IDictionary).
- No incluir directivas al precompilador.