Programación 4

EXAMEN DICIEMBRE 2017

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 examen junto al examen.
- Está prohibido el uso de computadoras, tabletas o teléfonos durante el parcial.

Problema 1 (35 puntos)

Una empresa que organiza eventos desea un sistema que le permita la gestión de los mismos. Los eventos se identifican con un código, tienen un costo fijo asociado y se desea registrar si son provistos directamente por la empresa o a través de una tercerización. De cada evento se conoce además su fecha de realización, la cantidad de asistentes y el lugar donde se realizará. Existen varios tipos de eventos más específicos: Familiares, Empresariales y Académicos. Para el caso de los Empresariales se registra el número de RUT de la empresa y su razón social; si es Académico, la institución y si es de carácter internacional o no.

Todo evento tiene un responsable de su organización, que trabaja en la empresa, y una persona contratante. Una persona contratante no puede ser a la vez responsable del mismo evento. Las personas son identificas por su cédula de identidad; además se conoce de ellas su nombre, apellido y fecha de nacimiento.

Un evento puede incluir varios servicios. Los mismos son de diferentes tipos: Conferencias, Ambientación del lugar, Servicio de catering, Recreación, Música. Las Conferencias serán dadas por uno o más expositores, personas de las cuales se desea conocer además su currículum vitae. Estos servicios no son ofrecidos para los eventos Familiares. Para el caso del Servicio de catering se desea conocer una descripción del menú y para el caso de Recreación una explicación sobre qué consiste el servicio.

Caso de Uso:	Registrar evento
Actor:	Administrador
Descripción:	Este caso de uso comienza cuando el administrador desea registrar un nuevo evento. Para ello debe ingresar los datos básicos correspondientes al evento, los específicos correspondientes al tipo de evento y los identificadores del contratante y del responsable del mismo. Luego debe ingresar los servicios asociados al evento, junto con la información correspondiente a cada uno.

Se pide:

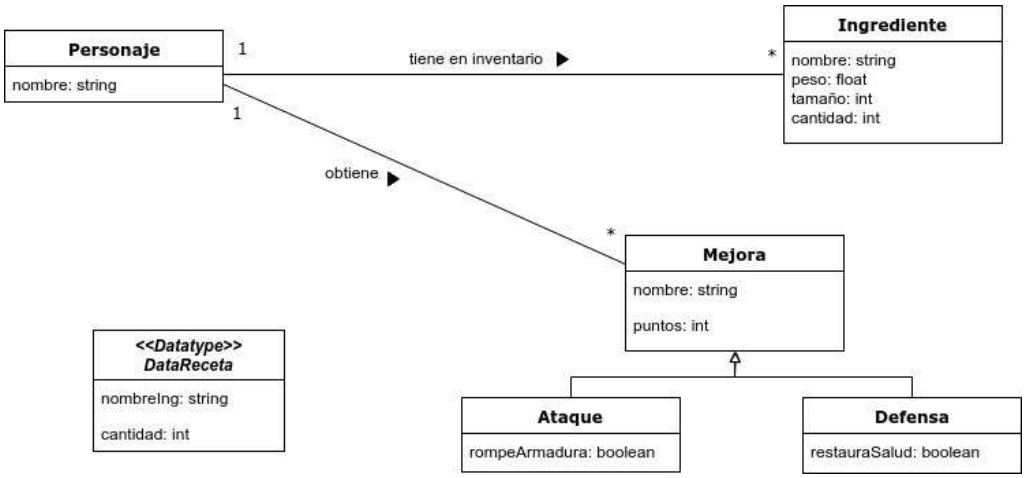
i) Modelo de Dominio de la realidad anterior (considerando el Caso de Uso) con restricciones en lenguaje natural.

ii) Diagrama de Secuencia del Sistema (DSS) del Caso de Uso, incluyendo el uso de Datatypes y de manejo de memoria del Sistema, en caso de ser necesario.

Problema 2 (35 puntos)

La empresa FinGames está desarrollando un nuevo videojuego, que se trata de un hechicero en la Tierra Media que sigue una historia y va creando mejoras que influyen en su desempeño, a lo largo de la misma. Como indica el modelo que se aprecia más abajo, el juego tiene distintos ingredientes (identificados por su nombre) que el personaje puede tener en su inventario y a su vez se sabe la cantidad que tiene de cada uno. Para crear una mejora que puede ser o bien de ataque o de defensa, el personaje consume diferentes ingredientes de su inventario.

Para cada mejora existe una receta, que es un conjunto con los nombres y las cantidades necesarias de cada ingrediente para crear dicha mejora. Por ejemplo, para crear la mejora de ataque *relámpago* se tiene la siguiente receta: *1 ojo de cíclope*, *3 cuernos de minotauro* y *1 alma de espectro*.



Parte I:

Se pide:

i) El juego tiene una única instancia del personaje principal. Indicar qué patrón conviene utilizar, y para dicho patrón explicar con sus palabras qué problema general resuelve, así como los atributos y operaciones básicas que debe tener la clase Personaje.
ii) Realice el Diagrama de Comunicación tomando en cuenta la parte anterior, incluyendo visibilidades y siguiendo los criterios GRASP, de la siguiente operación del sistema:.

`void crearMejora(nombreM:string, puntosM: int, receta: set<DataReceta>, esAtaque: boolean)`	
Descripción	Se crea una nueva mejora para el personaje a partir de una receta.
Parámetros	- *`nombreM`*: Nombre de la mejora a crear. - *`puntosM`*: Cantidad de puntos (de ataque o de defensa) que posee la mejora. - *`receta`*: Conjunto de *`DataReceta`*. - *`esAtaque`*: Indica si la mejora es de tipo ataque, si es falso, es de defensa.
Precondiciones	- El personaje no posee una mejora de nombre *`nombreM`*. - *`puntosM`* es un entero entre 1 y 100. - En el set de *`DataReceta`*, para cada elemento se cumple que: para el atributo *`nombreIng`* el personaje tiene un ingrediente con dicho nombre, y para el atributo *`cantidad`* tiene al menos esa cantidad de unidades del ingrediente correspondiente.
Postcondiciones	- Se crea una nueva instancia de *`Mejora`* con identificador *`nombreM`* y con puntos igual a *puntosM*. - Si *`esAtaque`* es true, la mejora es de clase *`Ataque`* y si *`puntosM`* > 80 entonces *`rompeArmadura`* es true (false en caso contrario). Si *`esAtaque`* es false la mejora es de clase *`Defensa`* y si *`puntosM`* > 90 entonces *`restauraSalud`* es true (false en caso contrario). - Se crea un link entre la nueva *`Mejora`* y el *`Personaje`*. - Para cada *`Ingrediente`* identificado por cada elemento de *`receta`*, se le resta *`cantidad`* al atributo *`cantidad`*. Si el atributo *`cantidad`* queda en 0, se destruye la instancia del *`Ingrediente`*.

Parte II:

Para el juego se tiene ahora una clase *`Menu`* que pertenece a la capa de presentación. Esta clase invoca de la capa lógica las operaciones del sistema que se piden en la parte I.

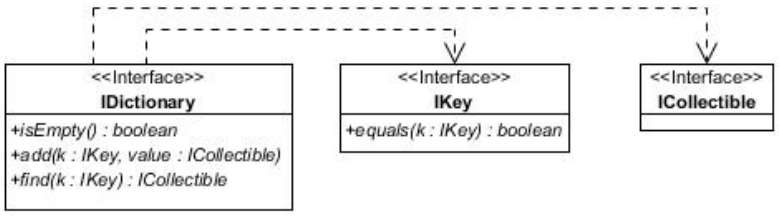
Se pide:

iii) Se desea que *Menu* **NO** conozca a ninguna a ninguna de las implementaciones de las clases del sistema. ¿Qué mecanismo puede utilizar para resolver el problema? Justifique su respuesta.
iv) Realice el Diagrama de Clases de Diseño (DCD) para soportar todo lo requerido en las partes anteriores (tanto de la parte I como de la II).

Aclaración: Incluir en el DCD únicamente los setters, getters, constructores y destructores que utilice en la parte I.

Problema 3 (30 puntos)

Se quiere ofrecer una realización (parcial) del diccionario genérico dado en el curso basada en una implementación de un árbol binario de búsqueda no balanceado (BST, por sus siglas en inglés). A continuación se recuerda lo que nos interesa para este problema, de las interfaces del Diagrama de Clases de Diseño (DCD) vinculadas al diccionario genérico.



Algunas consideraciones que nos interesan sobre la implementación:

- En los constructores se levanta la excepción ***std::invalid_argument*** cuando los valores de los parámetros no son los esperados (por ejemplo, *key* nula).
- La operación ***add*** levanta una excepción ***std::invalid_argument*** en caso de que ya exista una entrada en el diccionario con el mismo valor de *key*.
- La operación ***find*** devuelve ***null*** si el diccionario no contiene una entrada con el valor de *key* dado.
- Al destruir el diccionario, las *key* se destruyen pero no así los *value*.

Se pide:

i) Completar usando UML el DCD dado arriba que permita implementar la estructura de datos BST
ii) Implementar en c++ (sin usar la construcción *struct*) el DCD de la parte i que incluya:
    - las interfaces ***ICollectible***, ***IKey*** e ***IDictionary***,
    - las clases diseñadas para representar a la estructura BST incluyendo constructores y destructores,
    - las operaciones completas: ***isEmpty, add***,y ***find***.

Nota: No incluir directivas al precompilador.