

Programación 4

EXAMEN JULIO 2014

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 A:

Responda de forma breve, clara y precisa las siguientes preguntas:

- a) ¿Qué especifica un Contrato de Software?
- b) ¿Qué es una Clase en orientación a objetos?

Parte B:

Considere la siguiente descripción de la realidad a modelar.

Contexto

Los *videojuegos de rol multijugador masivos en línea* o MMORPG (siglas del inglés de *massively multiplayer online role-playing game*), son videojuegos que permiten a miles de jugadores introducirse en un mundo virtual de forma simultánea a través de internet. Estos han alcanzado una gran popularidad sobre todo por la posibilidad de interactuar con personas de todo el mundo, coordinando entre sí con el fin de cumplir un objetivo común.

Le han encomendado a usted y a su equipo realizar parte de la etapa de análisis de un nuevo videojuego, en el que los jugadores tendrán la posibilidad de utilizar héroes para el cumplimiento de distintas misiones.

Realidad

El juego está basado en la existencia de héroes, los cuales serán utilizados por los jugadores. De cada jugador interesa saber su nickname que lo identifica, su nombre, su apellido y su contraseña que utilizará para ingresar en el juego. Cada jugador puede tener hasta un máximo de tres héroes, mientras que un héroe pertenece a un único jugador.

De cada héroe interesa saber su nombre (que lo identifica), su puntaje de fuerza, su puntaje de destreza y su puntaje de inteligencia. Los héroes pueden tener una o más habilidades. Cada habilidad se identifica por un número, interesando además su descripción y si es de tipo ataque o de tipo defensa.

Los héroes pueden ser magos, duendes o humanos. De los magos interesa saber su puntaje de conocimientos mágicos y al mismo tiempo los hechizos que es capaz de

realizar. De cada hechizo interesa saber su nombre (que lo identifica) y la descripción del efecto que este posee.

De los humanos interesa saber una descripción de sus habilidades de pelea así como las armas que es capaz de utilizar. De cada arma interesa saber su nombre (que la identifica) y sus puntos de ataque. Un humano puede manejar una cantidad de armas menor o igual a la cantidad de habilidades que posee.

Los duendes pueden invocar hasta un máximo de tres criaturas. Las criaturas se identifican por su nombre, el cual está compuesto por el nombre de su héroe invocador como prefijo más caracteres adicionales. Al igual que los héroes las criaturas poseen habilidades, las cuales deben de ser distintas a las habilidades que posee el héroe que las invocó.

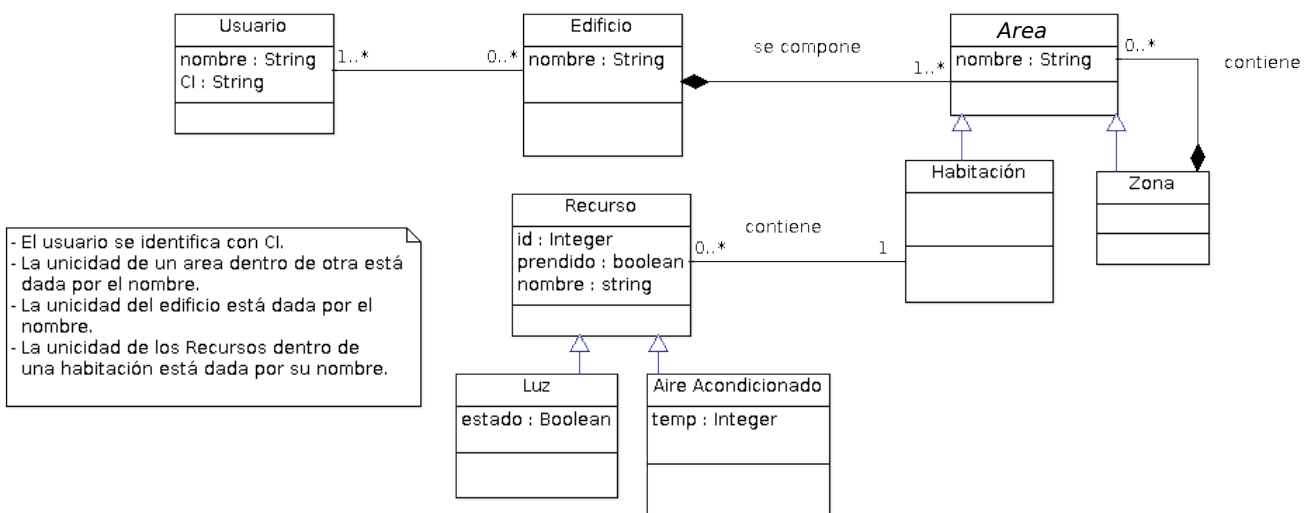
Por último, los jugadores pueden participar en misiones en conjunto con otros jugadores. Se debe tener en cuenta que cada jugador podría eventualmente usar más de un héroe en una misma misión. De cada misión interesa saber su número identificador, el objetivo y la descripción. Además es necesario tener un registro de todas las habilidades utilizadas por los héroes de cada jugador en cada misión.

Se pide:

- a) Modelar la realidad planteada mediante un Diagrama de Modelo de Dominio UML
- b) Expresar todas las restricciones del modelo en lenguaje natural.

Problema 2 (35 puntos)

P4LazyLife es una aplicación que permite manejar tu casa desde la comodidad de un pc o cualquier dispositivo móvil. En el programa pueden configurarse tantos edificios como se desee y asignarle usuarios a cada uno de ellos. Permite configurar su distribución espacial y los recursos contenidos en cada habitación. Cada habitación queda definida de forma global por su path (Ejemplo de path: 'PrimerPiso/Cocina/'). *P4LazyLife* cuenta con dos tipos de recursos, Luz y AireAcondicionado. Luz es capaz de prender y apagar, mientras que AireAcondicionado añade la opción de configurar una temperatura objetivo.



Parte A:

Se consideran los casos de uso *crearRecurso*, *prenderApagarRecurso* y *crearEdificio*, cada uno de los cuales es modelado con una sola operación del sistema, cuyos contratos se especifican a continuación.

crearRecurso(edificio: string, path: string, luz: boolean, nombre: string)	
Desc	Se crea un recurso de tipo luz o aireAcondicionado (instancia de luz si luz = true) en la ubicación path con el nombre nombre en el edificio edificio .
Pre	<ul style="list-style-type: none"> - No existe otro recurso con nombre nombre en la ruta path. - Existen las instancias de area y habitacion tal que la ruta path existe en el sistema para el edificio edificio.
Post	<ul style="list-style-type: none"> - Se crea una instancia de luz o aireAcondicionado, según corresponda. - Se crea un link entre la habitacion correspondiente al último tramo (Ultima zona del path) del path y el recurso creado, para el edificio edificio.

prenderApagarRecurso(edificio:string, path: string, nombre: string, on: boolean)	
Desc	Si on == true se prende el recurso en la ubicación path con el nombre nombre en el edificio edificio , si on == false se debe apagar.
Pre	No existe otro recurso con nombre nombre en la ruta path para el edificio edificio .
Post	Se asigna on al atributo prendido del recurso con nombre nombre , asociado a la habitación correspondiente al último tramo de la ruta path para el edificio edificio .

crearEdificio(nombre_e: string, nombre_a, ci_u: string)	
Desc	Se crea un Edificio con nombre nombre_e y se crea una zona con nombre nombre_a y se asocia al Usuario de cédula ci_u .
Pre	<ul style="list-style-type: none"> - No existe una instancia de Edificio con nombre nombre_e en el sistema. - Existe una instancia de Usuario con cédula ci_u en el sistema.
Post	<ul style="list-style-type: none"> - Se crea una instancia de Zona con nombre nombre_a. - Se crea una instancia de Edificio con nombre nombre_e. - Se asocian las instancias de Edificio y la instancia de Usuario con cédula ci_u. - Se asocian la nueva instancia de Edificio y la nueva instancia de Zona.

Se pide:

- a) Realizar el diagrama de comunicación para las operaciones previamente declaradas.
- b) Realizar el Modelo de Clase de Diseño correspondiente.

Parte B:

- a) ¿Qué es un patrón de diseño y cuál es su utilidad? Brinde un ejemplo de un patrón que no se haya usado en este ejercicio.
- b) ¿Qué patrones reconoce en su modelo de clase de diseño? Especifique los roles de cada clase.

Problema 3 (30 puntos)

Parte A:

Explique (a través de un ejemplo en C++) un escenario en el cual un espacio de memoria alojado en el heap queda inaccesible.

Parte B:

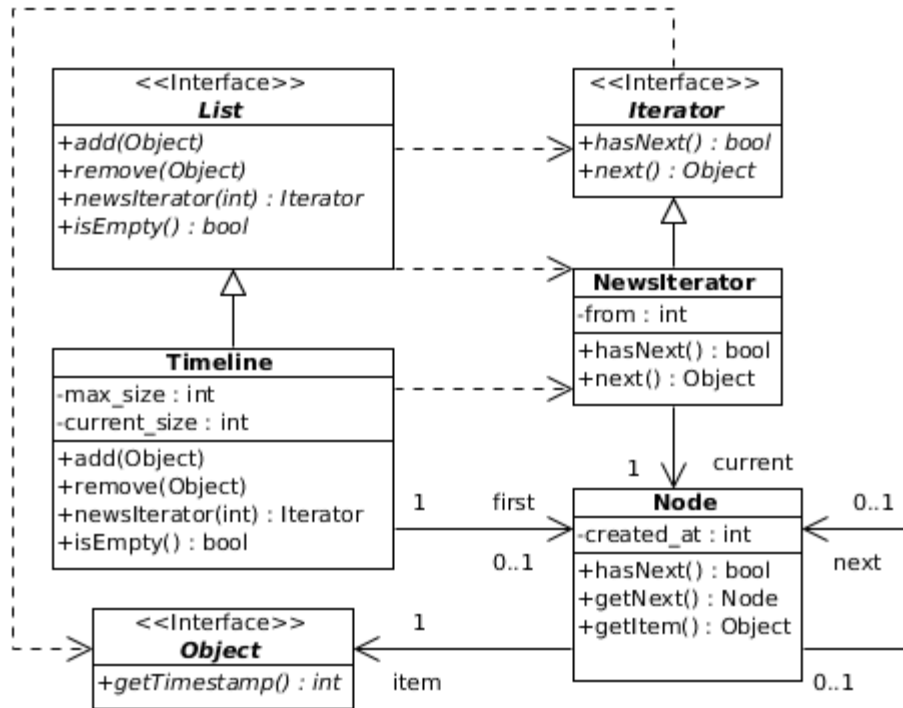
Se desea implementar una lista **Timeline** con las siguientes características:

- Cada vez que se agrega un nuevo elemento a la lista se desea mantener un timestamp que representa la fecha en la que el elemento fue creado (**created_at**).
- La lista va a tener un tope **max_size** (que debe ser configurable pero no va a cambiar luego de creada la lista). Si al agregar un nuevo elemento la cantidad de elementos supera el tope, se debe eliminar el elemento más antiguo.

A su vez, se desea contar con un iterador **NewsIterator** que permita recorrer los elementos creados luego de una fecha dada (*from*).

A partir de estos requerimientos se realizó el siguiente diseño de la solución:

(continúa)



Se pide:

Implementar completamente en C++ las clases **Timeline**, **Node**, **Iterator** y **NewsIterator**.

Notas:

- Se debe incluir (según corresponda) constructores, operaciones get/set y destructores.
- No es necesario incluir las directivas del preprocesador.
- Es posible comparar dos timestamps comparando los int que los representan.