

Tecnólogo Informático – San José

Programación Avanzada

Segundo Parcial 2016

- Completar **TODAS** las hojas con el nombre y el número de cédula.
- Escriba las hojas de un solo lado.
- No se puede utilizar material de ningún tipo. **Apagar celulares.**
- Solo se contestan dudas acerca de la letra de los ejercicios.
- El parcial dura 2 horas y media.

Problema 1 (35 puntos)

a) Explique para qué sirven los criterios GRASP y describa 3 de ellos brevemente.

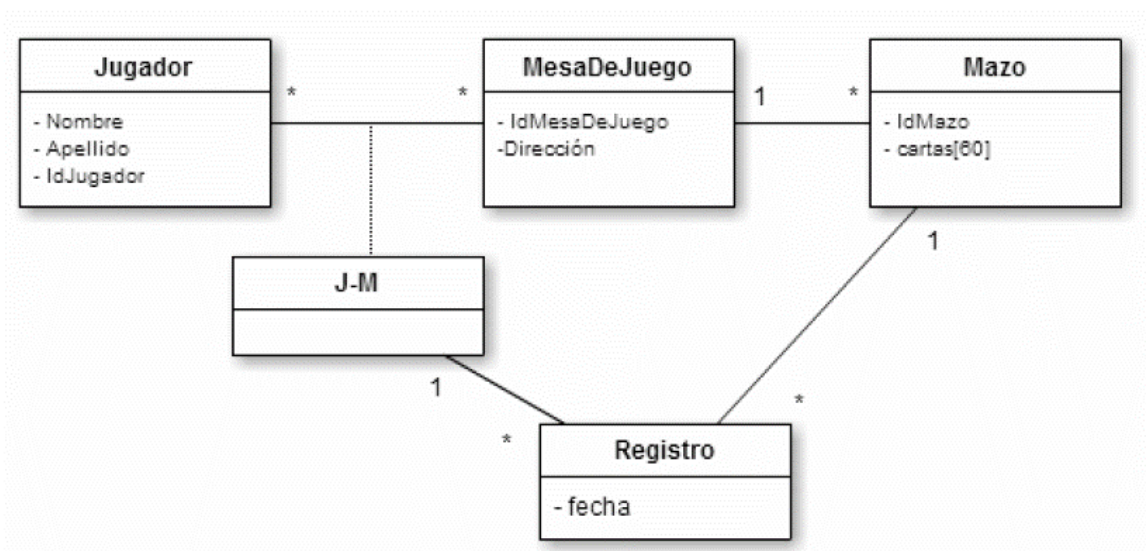
b) Contexto:

Magic: el encuentro, originalmente en inglés **Magic: The Gathering** es un juego de cartas coleccionables. Cada partida de Magic representa una batalla entre dos jugadores, en donde cada jugador posee un mazo de cartas coleccionables que son usadas para la batalla. Un sistema organizado de torneos y una comunidad de jugadores profesionales se ha desarrollado alrededor del juego, por lo que a usted y a su equipo de desarrollo le han encomendado la tarea de diseñar un pequeño módulo que tiene como finalidad llevar registro de un torneo magic, llamado MagicUT (Magic Uruguayan Tournament).

Realidad:

En MagicUT existe lo denominado “mesa de juego” que es una instancia en la que varios jugadores se reúnen en un lugar acordado y compiten entre sí. En el torneo cada jugador puede haber participado en múltiples mesas de juego, así como cada una de estas puede haber incluido a varios jugadores. Interesa registrar las fechas en las que el jugador participó en cada mesa, así como el mazo que utilizó. Cada mesa de juego puede poseer varios mazos pre-armados (los cuales serán utilizados por los jugadores que participen en dicha mesa), así como cada mazo pertenece a una única mesa de juego.

El equipo de Análisis le ha otorgado el siguiente modelo:



Restricciones:

- 1) Un jugador se identifica por IdJugador
- 2) El concepto MesaDeJuego se identifica por IdMesaDeJuego
- 3) Cada mazo se identifica por IdMazo
- 4) Un jugador que genera un registro de un mazo en una mesa de juego, participó en esa mesa de juego.

Se considera el caso de uso “Agregar nuevo registro a jugador en mesa de juego”, el cual es modelado con una única operación del Sistema, cuyo contrato se especifica a continuación:

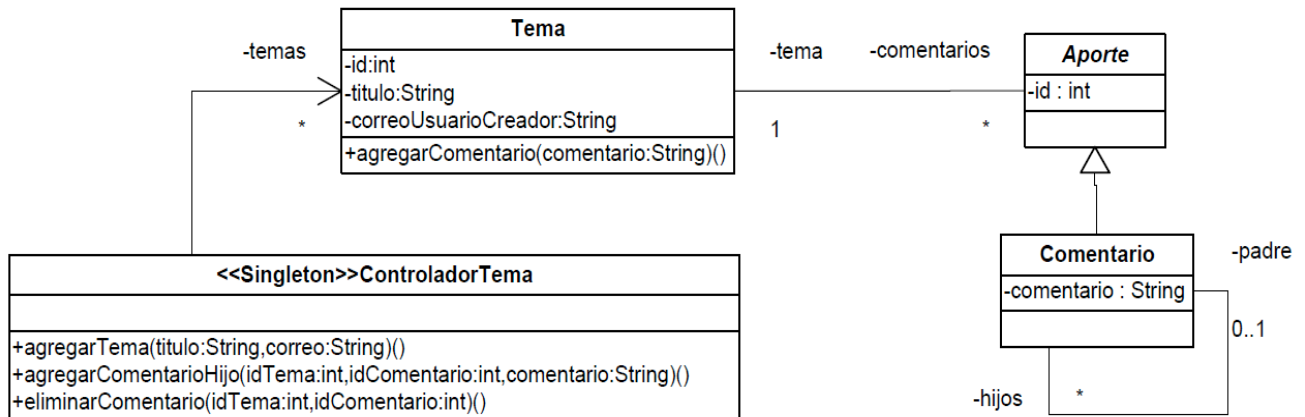
registrarJugadorEnMDJ(IdJ, IdMDJ, IdMazo: Integer, fActual: DateTime)	
Descripción	Se agrega un nuevo registro de un jugador en una mesa de juego en la que ya ha participado previamente
Parámetros	<ul style="list-style-type: none">- IdJ: Id del jugador.- IdMDJ: Id de la mesa de juego.- IdMazo: Id del mazo.- fActual: fecha actual del Sistema.
Precondiciones	<ul style="list-style-type: none">- Existe en el Sistema una instancia de Jugador con IdJugador igual a IdJ.- Existe en el Sistema una instancia de MesaDeJuego con IdMesaDeJuego igual a IdMDJ.- Existe en el Sistema una instancia de Mazo con IdMazo igual a IdMazo.- Existe una instancia de tipo asociativo J-M entre un Jugador cuyo IdJugador es igual a IdJ y una MesaDeJuego cuyo IdMesaDeJuego es igual a IdMDJ- Existe una asociación entre una instancia de Mazo cuyo IdMazo es igual a IdMazo y una instancia de MesaDeJuego cuya IdMesaDeJuego es igual a IdMDJ
Postcondiciones	<ul style="list-style-type: none">- Existe en el Sistema una nueva instancia de Registro con fecha actual.- Se crea una asociación entre el nuevo Registro y el tipo asociativo J-M de Jugador cuyo IdJugador es igual a IdJ y MesaDeJuego cuyo IdMesaDeJuego es igual a IdMDJ.- Se crea una asociación entre el nuevo Registro y el Mazo cuyo IdMazo es igual a IdMazo.

Se pide:

- a) Realizar el Diagrama de Comunicación para la operación previamente declarada.
- b) Realizar el Diagrama de Clases de Diseño correspondiente.

Problema 2 (25 puntos)

- a) Se dispone del siguiente modelo de una parte de un sistema donde se pueden ingresar temas de interés, y hacer comentarios sobre los mismos.



Se pide:

- a) Implemente los `.h` de todas las clases que aparecen en el modelo, según corresponda con sus atributos, asociaciones, operaciones, constructores y destructores.
- b) Implemente las siguientes operaciones de la clase **ControladorTema**:
- **eliminarComentario**: elimina la instancia de Comentario con id igual a idComentario del tema idTema
 - **agregarTema**: crea un nuevo Tema con el título y correo del usuario creador pasados por parámetro (el id es autogenerado)
 - **agregarComentarioHijo**: agrega un comentario a la colección de hijos del comentario con idComentario, dentro del Tema con idTema. (el id del nuevo comentario es autogenerado)

Observaciones generales:

- No incluir directivas al pre compilador.
- Puede suponer la existencia de implementaciones de IDictionary, ICollection, IEnumerator y KeyInt según sea necesario.
- Todos los atributo denominados como id son autogenerados, asuma que existe una clase denominada **Util**, que puede utilizar y que dispone de una operación estática: `int getId()`, que devuelve un entero distinto cada vez que es invocada.
- En la eliminación de elementos, todos los borrados son en cascada, es decir, para un tema se eliminan todos sus comentarios, para un comentario, todos los comentarios sobre sí mismos.