Tecnólogo Informático – San José

Programación Avanzada

Segundo Parcial 2019

- *Completar todas las hojas con el nombre y el número de cédula.*
- *Numerarlas y escribir el total en la primer hoja. Escriba las hojas de un solo lado.*
- *No se puede utilizar material de ningún tipo.* ***Apagar celulares.***
- *Sólo se contestan dudas acerca de la letra de los ejercicios.*
- *El parcial dura 2 horas y media.*

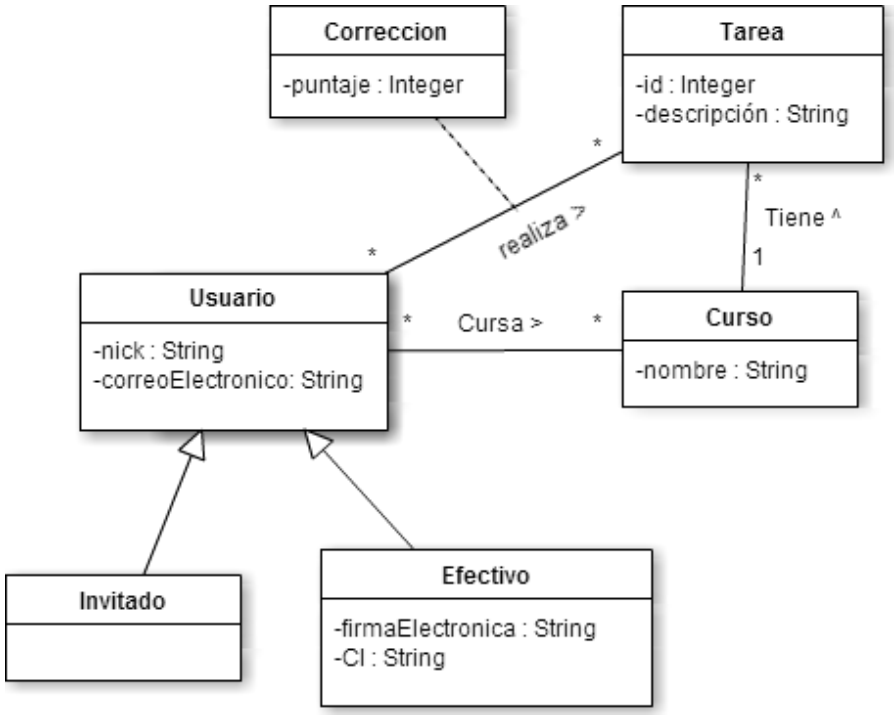
Ejercicio 1 (30 puntos)

A usted y a su equipo de desarrollo se le ha encomendado la tarea de diseñar un módulo específico de un Sistema que permite la realización de cursos online por parte de usuarios.

Básicamente los Usuarios pueden ser de dos tipos: Invitado y Efectivo. Todos los usuarios pueden inscribirse a los cursos (esto es: acceder al material, realizar las Tareas, etc) con la única diferencia de que los usuarios efectivos tendrán diploma y los invitados no. Por ese motivo se desea conocer de los efectivos una firma electrónica y su cédula de identidad.

Los cursos poseen un conjunto de tareas que los usuarios deberán realizar. El Sistema debe mantener registro de los puntajes obtenidos por los usuarios en cada tarea.

A continuación se muestra un modelo de dominio obtenido en la etapa de análisis:



Restricciones:

- Los Usuarios se identifican por el nick.
- Las Tareas se identifican por el id dentro del curso únicamente.
- Los Cursos se identifican por el nombre.
- Un Usuario realiza Tareas pertenecientes a un Curso al que está inscripto.

Se consideran los casos de uso "Agregar nueva corrección a Usuario" y "Obtener información de un Curso", cada uno de los cuales es modelado con una única operación del Sistema, cuyos contratos se especifican a continuación:

agregarNuevaCorreccion (nick : String, nombreCurso : String, idTarea: Integer, puntaje: Integer)	
Descripción	Se agrega un nueva Corrección al Usuario identificado con el nick pasado por parámetro en una Tarea realizada por él.
Parámetros	- nick: Nick del Usuario. - nombreCurso: nombre del Curso. - idTarea: Id de la Tarea a la que se le asignará la corrección. - puntaje: Puntaje obtenido por el Usuario en la Tarea.
Pre-condiciones	- Existe en el Sistema una instancia de Usuario con nick igual a nick. - Existe en el Sistema una instancia de Curso con nombre igual a nombreCurso. - Existe en el Sistema una instancia de Tarea con id igual a idTarea asociada a un Curso con nombre igual a nombreCurso. - **NO** existe una instancia de tipo asociativo Corrección entre un Usuario con nick igual a nick, y una Tarea cuyo id es igual a idTarea asociada a un Curso con nombre igual a nombreCurso. - Existe un link entre una instancia de Usuario cuyo nick es igual a nick y una instancia de Curso cuyo nombre es igual a nombreCurso.
Post-condiciones	- Existe en el Sistema una nueva instancia de tipo asociativo Corrección con puntaje igual a puntaje, entre la instancia de Curso cuyo nombre es igual a nombreCurso y la instancia de Usuario cuyo Nick es igual a nick

obtenerInformaciónDeCurso(nombreCurso : String): DataCurso	
Descripción	Se obtiene información de un Curso
Parámetros	- nombreCurso: nombre del Curso del que se desea obtener información.
Pre-condiciones	- Existe en el Sistema una instancia de Curso con nombre igual a nombreCurso.
Post-condiciones	- Se retorna un datavalue con la siguiente información: o Nombre del Curso cuyo nombre sea igual a nombreCurso. o Nick y correo electrónico de todos los usuarios inscriptos al curso cuyo nombre es igual a nombreCurso. En caso de que sean Usuarios efectivos se retorna su firma electrónica y su C.I. o Id y descripción de todas las Tareas asociadas al curso cuyo nombre es igual a nombreCurso.

Se pide:

i. Realizar el Diagrama de Comunicación para cada una de las operaciones declaradas.

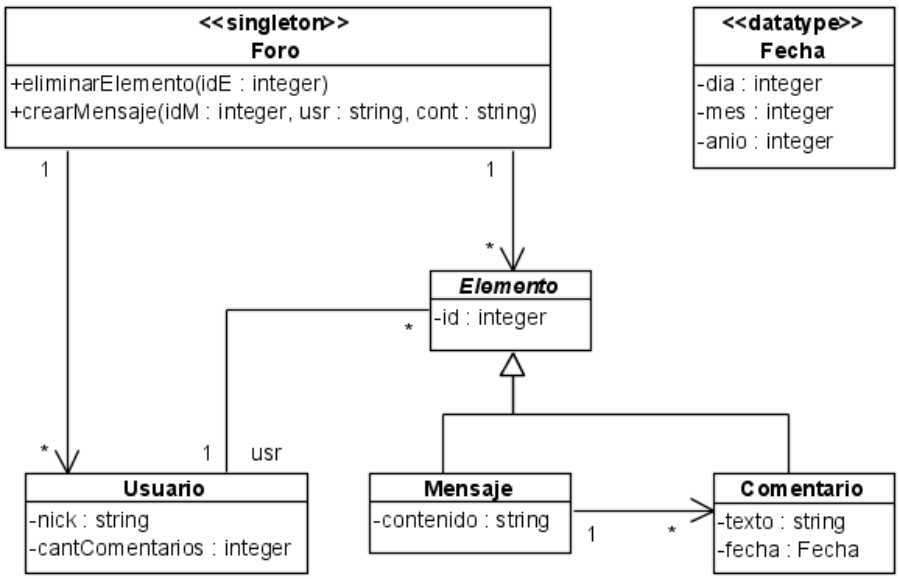
ii. Realizar el Diagrama de Clases de Diseño.

Ejercicio 2 (30 puntos)

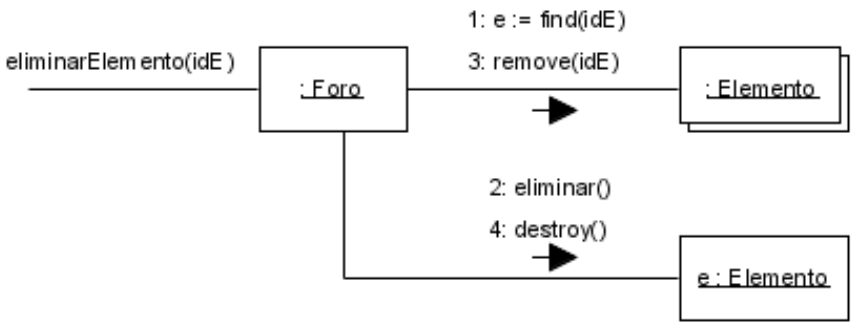
Se le ha encomendado la implementación de un foro de mensajes que se utilizará como un componente durante la construcción de un sitio web donde los usuarios pueden compartir información interesante como links a páginas web u otros contenidos de interés. Para esto el sistema mantiene una lista de usuarios activos que pueden agregar nuevos mensajes en el foro y comentar sobre ellos. De cada mensaje se conoce su contenido y los comentarios de los usuarios. De cada usuario se conoce su nick (que lo identifica) y la cantidad de comentarios que realizó. De cada comentario se conoce su texto y la fecha en la cual fue realizado.

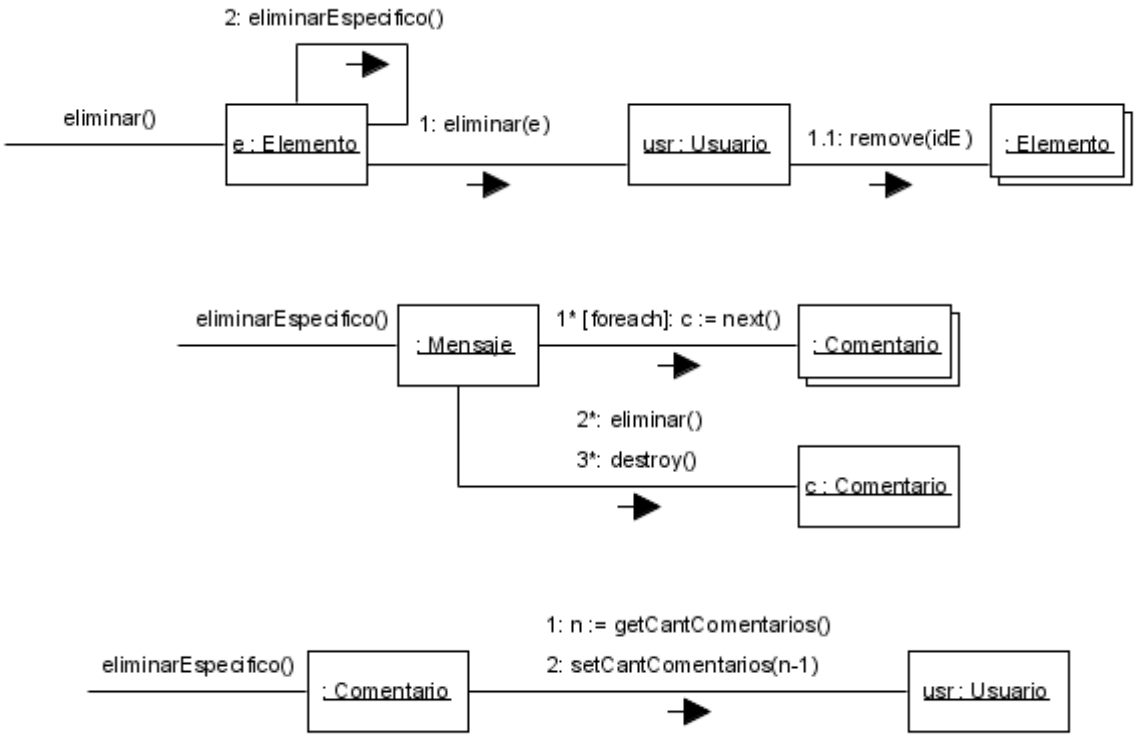
Por una decisión tomada en la etapa de diseño, tanto los mensajes como los comentarios se identifican por un id, es decir, no pueden existir dos mensajes o comentarios con el mismo id. Para cada usuario se almacena los comentarios y mensajes que ingresó en el foro.

El equipo de desarrollo generó el siguiente diagrama de clases **parcial** que se deberá seguir en la implementación.



Es necesario dar una implementación para la operación ***eliminarElemento()*** que permite quitar un mensaje o un comentario del foro. Si es un mensaje, se eliminan todos los comentarios asociados, y si es un comentario, se actualiza el valor de la variable cantComentarios del usuario que generó ese comentario. A continuación se brinda un diagrama de comunicación para la operación.





Se pide:

i. Escribir el código de los **.h** de las clases **Foro**, **Elemento**, **Mensaje** y **Comentario**, no incluya getters ni setters.

ii. Implemente el método ***eliminarElemento*** de la clase **Foro** (**.cpp**), incluyendo la implementación de las operaciones relacionadas, correspondientes a las clases **Usuario**, **Elemento**, **Mensaje** y **Comentario**, que sean invocadas por el método mencionado.

Notas:

- No incluya directivas de preprocesador
- Asuma que existe una implementación estándar de las interfaces ICollectible, ICollection, IIterator, IDictionary e IKey.
- Asuma que existe una clase **List** que realiza la interfaz ICollection y una clase **ListDictionary** que realiza la interfaz IDictionary.
- Además suponer que existen las clases **KeyString** y **KeyInteger** que implementan IKey y permiten almacenar claves de tipo String y entero respectivamente.