Introducción a la Programación – UNQ – 2do semestre de 2012

Segundo parcial – Blog de Noticias

Aclaraciones:

- *Esta evaluación es a libro abierto. Se pueden usar todo lo visto en la práctica y en la teórica, aclarando la referencia.*
- *No se olvide de poner nombre, nro. de legajo, nro. de hoja y cantidad total de hojas en cada hoja.*
- *Le recomendamos leer el enunciado en su totalidad y organizar sus ideas antes de comenzar la resolución.*
- *Recuerde que la intención es medir cuánto comprende usted del tema. Por ello, no dude en escribir todo lo que sabe, en explicar lo que se propone antes de escribir código, en probar su código con ejemplos, etc.*

El presente parcial tiene por finalidad simular la operatoria de un *Blog de Noticias*. Este blog está constituido por noticias y lectores. Los lectores pueden leer las noticias y calificarlas de 1 a 10 (indicando cuánto les gustó – 1 es nada y 10 es que le encantó). Las noticias también tienen noticias relacionadas que invita a los lectores a seguir leyendo sobre temas relacionados.

Para modelar el mencionado blog, se introducen las estructuras de datos que a continuación se describen. Una *noticia* consiste de un identificador, el contenido neto de la noticia[1], una lista de calificaciones y una lista de los identificadores de noticias relacionadas. Una *calificación* está dada por una identificación de lector (aquel que realizó la calificación) y la calificación misma. El *lector* está dado por su identificación y la lista de identificadores de noticias que leyó. Finalmente un *blog de noticias* consiste en una lista de noticias, una lista de lectores, un identificador para ser utilizado al dar de alta a una nueva noticia (más detalles en el ejercicio 1) y un identificador para ser utilizado al dar de alta a un nuevo lector.

```
struct Noticia
{
    int idN;
    int contenido;
    List<Calificacion> cs;
    List<int> noticiasRelacionadas;
}

struct Calificacion
{
    int idL;
    int calificacion; // de 1 al 10
}

struct Lector
{
    int idL;
    List<int> noticiasLeidas;
}

struct BlogDeNoticias
{
     List<Noticia> ns;
     List<Lector> ls;
     int idProxNoticia;
     int idProxLector;
};
```

[1] Debería ser una cadena de caracteres pero para simplificar usamos números.

Se solicita resolver los siguientes ejercicios. Tener en cuenta que *ninguno* de ellos involucra el tablero de CGobstones.

Ejercicio 1 `BlogDeNoticias altaDeNoticia (BlogDeNoticias nb, int contenido, List<int>noticiasRelacionadas)`

Propósito: *Da de alta una nueva noticia. La identificación asignada a la misma está dada por el valor del campo* `idProxNoticia`. *Este campo debe ser incrementado una vez que el alta se efectiviza, de modo tal que la siguiente noticia que se agregue no repita ese número de identificación.*

Precondición: *No existe una noticia en* `nb` *con identificación* `idProxNoticia`.

Ejercicio 2 `int noticiaConMasCalificaciones (BlogDeNoticias nb)`

Propósito: *Retorna la identificación de la noticia que tiene más cantidad de calificaciones.*

Precondición: *Hay al menos una noticia en el blog.*

Ejercicio 3 `int noticiaMasLeida (BlogDeNoticias nb)`

Propósito: *Retorna la identificación de la noticia más leída por todos los lectores del Blog de Noticias.*

Precondición: *Hay al menos una noticia en el el blog.*

Ejercicio 4 `bool noticion (BlogDeNoticias nb, int idNoticia)`

Propósito: *Retorna verdadero si* `idNoticia` *es un* notición*. Un notición es una noticia que tiene calificación 9 o 10 y tal que* todas *sus noticias relacionadas también tienen calificaciones 9 o 10.*

Precondición: *La noticia* `idNoticia` *existe y tiene al menos una calificación.*

Ejercicio 5 `bool esGuiaDeLectura (BlogDeNoticias nb, List<int>guia)`

Propósito: *Retorna verdadero si* `guia` *es una guía de lectura válida. Una guía de lectura es una lista de identificaciones de noticias tal que si* `id2` *le sigue a* `id1` *en la lista entonces* `id2` *es una noticia relacionada con* `id1`*.*

Precondición: *Las identificaciones de noticias de* `guia` *existen en nb y además no tiene identificaciones repetidas.*

Ejercicio 6 `bool calificadorSensato (BlogDeNoticias nb, int idL)`

Propósito: *Retorna verdadero si* todas *las calificaciones realizadas por el lector* `idL` *para cada noticia difieren en a lo sumo un punto del promedio de las calificaciones de esa noticia.*

Precondición: *El lector* `idL` *existe y calificó al menos una noticia.*

Puede asumir la existencia de `promedioDeCalificaciones` *que dada una lista de calificaciones retorna el promedio. También puede hacer uso de* `lectorCalificoNoticia` *que dado una identificación de lector y una noticia retorna un booleano indicando si ese lector calificó esa noticia o no.*