

UNIDAD TEMÁTICA 4: Árboles Binarios I

TRABAJO DE APLICACIÓN 9

Escenario

Una institución de asistencia médica registra a sus cientos de miles de afiliados en un árbol binario de búsqueda, llamado `padronAfiliados`. El registro y la identificación del afiliado se hace a partir de la cédula de identidad (8 cifras numéricas, sin puntos ni guiones).

Cada afiliado, además de los datos personales básicos (cédula de identidad y nombre y apellido), tiene una lista con el histórico de consultas médicas realizadas y otra lista de consultas médicas a las que está anotado.

Los datos de una consulta son: fecha (de tipo fecha), especialidad (string), médico (número de cédula) y resultado. Resultado es un atributo de tipo entero que puede tener tres valores: 0 si todavía no se ha producido, 1 si el afiliado concurreó y 2 si el afiliado no concurreó, y 3 si la consulta ha sido cobrada. Al realizarse la anotación, el campo se inicializa en 0, y a la hora de la consulta, el personal de la policlínica registra en 1 si el afiliado concurre, o en 2 si cerrada la hora de la consulta el afiliado no concurreó.

El gerente de aplicaciones ha encargado el desarrollo de las siguientes funcionalidades:

- para cada afiliado, remueva de la lista las consultas pendientes del día anterior y las inserte en la lista histórica de consultas pasadas: `void actualizarConsultas()`
- devuelva un árbol binario de los afiliados que, estando anotados, no hayan concurrido a su consulta el día anterior: `TArbolBB<Afiliado> listadoContaduria()`
- devuelva un árbol binario de búsqueda de todos los afiliados que no tienen ninguna consulta, ni agendada, ni histórica: `TArbolBB<Afiliado> listadoInactivos()`
- elimine del padrón a todos aquellos afiliados que no hayan concurrido a más de una consulta: `TLista<Afiliado> limpiar()`

Consigna

Los equipos han de subdividirse en dos subequipos y trabajarán separados e independientes:

- Crearán el proyecto `UT4_TA9` a partir de las clases e interfases trabajadas en clase.
- Crear las interfaces y clases de afiliado y de consulta, y un `TArbolAfiliados` que extienda el `TArbolBB<Afiliado>` y en donde se escribirán los métodos a desarrollar.
- Ejecutarán la aplicación a partir de los archivos que se liberarán, para ello deben instanciar y cargar el padrón, y realizar las consultas indicadas.
- Emitirán por consola el resultado de las consultas
- Remitirán, no después de las 19:45 a la tarea de la webasignatura, un archivo **.ZIP** con los nombres de los integrantes del subequipo como nombre, conteniendo el proyecto.
- Se intercambiarán las soluciones y sincronizarán en GIT un único proyecto.