Prácticas de Estructuras de Datos y de la Información.

Práctica 4. La clase Lista

Objetivos

El principal objetivo de esta práctica es contrastar el TAD Lista (de enteros con cursor) en C++ que vamos a hacer aquí con la lista sin TAD y en C del curso pasado.

Enunciado

Diseñar el TAD Lista de enteros con cursor con la siguiente interfaz:

1. tama(): Número de elementos de la lista.
2. vacia(): Comprueba si una lista se encuentra vacía.
3. reset(): Situar el cursor de la lista en el primer elemento, si existe.
4. siguiente(): Situar el cursor de la lista en el siguiente elemento, si existe.
5. get(): Observador del elemento sobre el que se encuentra el cursor.
6. set(): Modificador del elemento sobre el que se encuentra el cursor.
7. insertaPrincipio(): Inserción de un elemento al principio de la lista. El cursor queda en el elemento insertado.
8. borra(): Borrado del elemento sobre el que se encuentre el cursor. El cursor queda en el elemento siguiente.
9. El constructor inicializará la Lista y el destructor debe borrar todos los elementos de la lista y liberar la memoria utilizada por ésta.

Se deben considerar los siguientes aspectos:

- Se creará una clase en C++ denominada Lista, que mantenga tres punteros privados: `_cabeza`, que apunta al comienzo de la lista, `_cola` que apunta al fin de la lista, y `_cursor` que apunta al elemento actual de la lista. Los tres se inicializan en el constructor.
- La clase Lista mantendrá un entero (`_t`) que en todo momento almacenará el número de elementos en la lista.
- La definición de la clase irá en el fichero `lista.h` y el cuerpo de las funciones en el fichero `lista.cpp`.
- Se realizará un programa de prueba (`menulista.cpp`), que desde la consola permitirá acceder a cada una de las operaciones anteriores además de permitir visualizar todos los elementos de la lista en pantalla para comprobar el estado de la lista cuando se desee.