

Fundamentos de la programación II

Laboratorio. Hoja 7

Los ejercicios marcados como **Evaluable** tienen que subirse al CV durante la sesión de laboratorio.

- Sólo hará la entrega uno de los miembros del grupo.
- La línea 1 (y siguientes) deben contener los nombres de los alumnos que hacen la entrega de la forma:

```
// Nombre Apellido1 Apellido2
// Nombre Apellido1 Apellido2
```

-
1. Consideremos la clase de las listas enlazadas de enteros vista en clase (disponible en el CV). En este ejercicio se trata de enriquecerla con los nuevos métodos que se especifican a continuación. Para cada uno de ellos deberá incluirse la opción correspondiente de menú en el programa principal, para probar la funcionalidad.

Los métodos pedidos son (algunos ya implementados):

- `public void InsertaIni(int e)`: inserta el elemento `e` al principio de la lista.
- `public int Suma()`: devuelve la suma de los elementos de la lista.
- `public int CuentaEltos()`: devuelve el número de elementos de la lista.
- **(Entregable)** `public int CuentaOcurrencias(int e)`: devuelve el número de ocurrencias del elemento `e` en la lista.
- `private Nodo NesimoNodo(int n)`: devuelve el `n`-ésimo nodo de la lista; `null` si no existe tal elemento.
- `public int Nesimo(int n)`: utilizando el método anterior, devuelve el `n`-ésimo elemento de la lista; lanza una excepción en si no existe tal elemento.
- `public void InsertaNesimo(int n, int e)`: inserta el elemento `e` en el lugar `n`-ésimo de la lista (puede ser útil el método `Nesimo`).
- `public bool BorraElto(int e)`: si el elemento `e` está en la lista borra su primera aparición y devuelve `true` en ese caso; deja la lista intacta y devuelve `false` si `e` no está en la lista.
- `public void BorraTodos(int e)`: borra todas las apariciones del elemento `e` en la lista.
- `public void BorraNesimo(int n)`: utilizando `NesimoNodo` borra el `n`-ésimo elemento de la lista, si existe.
- `public void Invierte()`: invierte el orden de los elementos de la lista.
- `public bool Iguales(Lista l)`: comprueba si la lista del objeto actual (`this`) es igual a `l`, i.e., tiene los mismos elementos en el mismo orden.
- `public bool Iguales2(Lista l)`: comprueba si la lista del objeto actual (`this`) tiene los mismos elementos que `l` aunque sea en distinto orden.

2. Consideremos ahora otra clase para las listas enlazadas enriquecida con `ult` (referencia al último elemento de la lista) y `numElems` (número de elementos de la lista). Implementar esta nueva clase con los métodos del ejercicio anterior, mejorando la eficiencia en los métodos donde sea posible. Gestionar los posibles errores de los métodos implementados mediante excepciones.