

## RETO 4: Arboles

Este reto consta de 2 partes.

- Diseñar un procedimiento inorden no recursivo a imagen y semejanza del procedimiento preorden no recursivo que yo diseñé en clase:

---

```
void preordenNR(const ArbolBinario<int>& a)
{
    ArbolBinario<int>::Nodo actual;
    stack<ArbolBinario<int>::Nodo> p;

    actual= ArbolBinario<int>::nodo_nulo;
    p.push(actual);

    actual= a.raiz();
    while (actual!=ArbolBinario<int>::nodo_nulo) {
        cout << a.etiqueta(actual) << \' \' ;
        if (a.derecha(actual)!=ArbolBinario<int>::nodo_nulo)
            p.push(a.derecha(actual));
        if (a.izquierda(actual)!=ArbolBinario<int>::nodo_nulo)
            actual= a.izquierda(actual);
        else {
            actual=p.top();
            p.pop();
        }
    }
}
```

---

- Dar un procedimiento para guardar un árbol binario en disco de forma que se recupere la estructura jerárquica de forma unívoca usando el mínimo número de centinelas que veais posible. Tendrá puntuación doble el que consiga dar la mejor solución que será la que use un menor número de datos para el proceso de salvado a disco del árbol.

### **Consideraciones:**

1. Este reto se puede hacer por parejas. La solución deberá introducirse en el sistema a través de un fichero pdf. Se sugiere como nombre del fichero reto4.pdf
2. Todas las soluciones correctas se puntuarán con 0.2
3. Me interesa analizar vuestra capacidad de abstracción, de forma que los algoritmos que os pido que hagais no espero que sean infinitamente precisos y que de ahí salga directamente el código para implementarlos. Espero una secuencia de pasos inteligente que resuelva el problema.
4. El plazo para entregar vuestras soluciones (que debereis dar obligatoriamente en un fichero pdf) se extenderá hasta las 23.55h del Lunes 16 de Diciembre.