

Actividad 6. Implementación del TAD Árbol General

Objetivo

Implementar el TAD árbol general haciendo uso de una estructura enlazada.

Procedimiento

1. Consultar las transparencias sobre árboles generales, en concreto, la sección de implementación, que están disponibles en **Tema**, sección **Documentos e Ligazóns / Teoría / ÁrbolGeneral.pdf**
2. Resolver los ejercicios que se indican en esta actividad, utilizando el lenguaje java. Para probar su correcto funcionamiento se puede hacer uso de los tests disponibles en Tema, sección Documentos e Ligazóns / Actividades / Test / **Actividad6Ejercicio1Test.java y Actividad6Ejercicio2Test.java.**

Evaluación

Estos contenidos serán evaluados mediante una prueba *individual* el 3 de noviembre.

Tiempo estimado

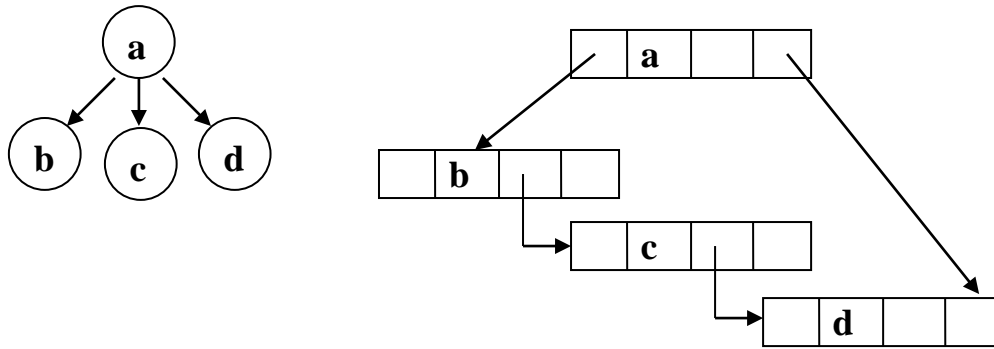
4 horas

Ejercicio

Una de las formas más habituales de implementar un árbol general es mediante un árbol binario, en el que el hijo izquierdo del árbol binario es el hijo más a la izquierda del árbol general y el hijo derecho del árbol binario es el hermano derecha del árbol general.

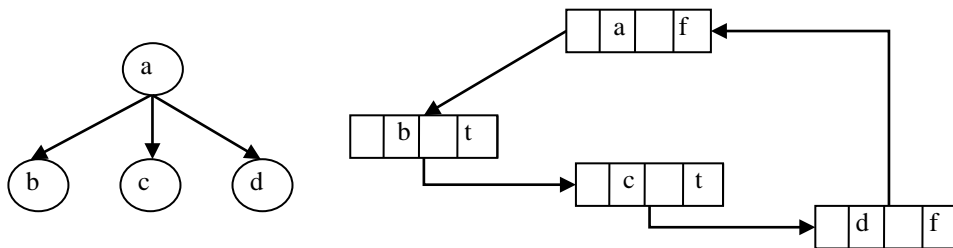
Para optimizar la implementación de algunas operaciones con árboles generales, existen estructuras alternativas:

- 1.- Si se necesita conocer con frecuencia cuál es el último hijo (hijo más a la derecha) de un determinado elemento, una solución consiste en usar una referencia extra en cada nodo para apuntar a su hijo más a la derecha.



Teniendo en cuenta esta nueva alternativa de implementación, se pide crear un proyecto que implemente el TAD Árbol General. Ten presente que la clase `NodoGeneral<E>` ha cambiado y que ahora incluye un nuevo atributo.

2.- Si se necesita conocer con frecuencia cuál es el padre de un determinado elemento, una solución consiste en usar la referencia hermano que está a null en el último hijo para apuntar al nodo padre. En este caso también necesitaríamos un atributo extra (boolean) en cada nodo para indicar si hemos llegado al final en la lista de hijos o no.



Teniendo en cuenta esta nueva alternativa de implementación, se pide crear un proyecto que implemente el TAD Árbol General. Ten presente que la clase `NodoGeneral<E>` ha cambiado y que ahora incluye un nuevo atributo (variable de objeto) de tipo boolean.

Anexo:

• TAD Árbol General:

```
public interface ArbolGeneral<E>{
    public boolean esVacio();
    public E raiz() throws ArbolVacioExcepcion;
    public ArbolGeneral<E> hijoMasIzq() throws ArbolVacioExcepcion ;
    public ArbolGeneral<E> hermanoDer() throws ArbolVacioExcepcion;
    public boolean esta(E elemento);
    public void setRaiz(E elemRaiz) throws ArbolVacioExcepcion;
    public void setHijo(ArbolGeneral<E> hijo) throws ArbolVacioExcepcion, NullPointerException;
    public void suprimir();
}
```