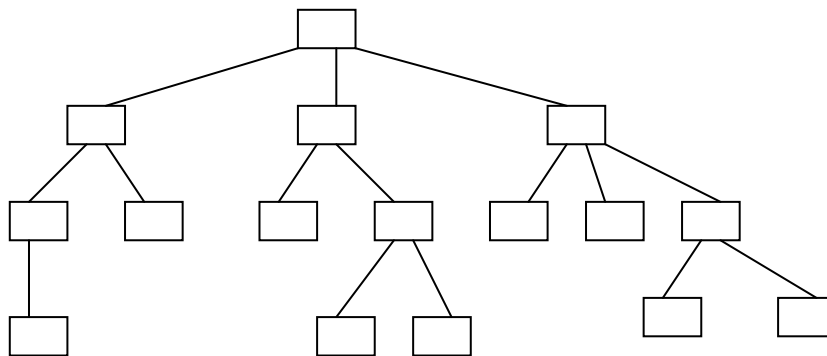
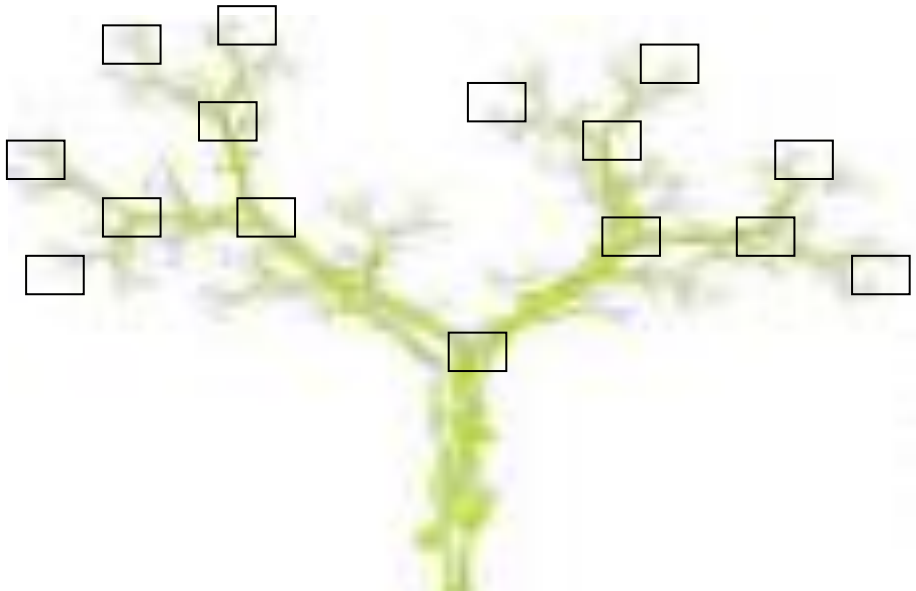


## ÁRBOLES

Para decidir que tipo de estructura es un árbol, basta con observar un árbol en la naturaleza, ubicando un nodo en donde cada rama inicia o termina.



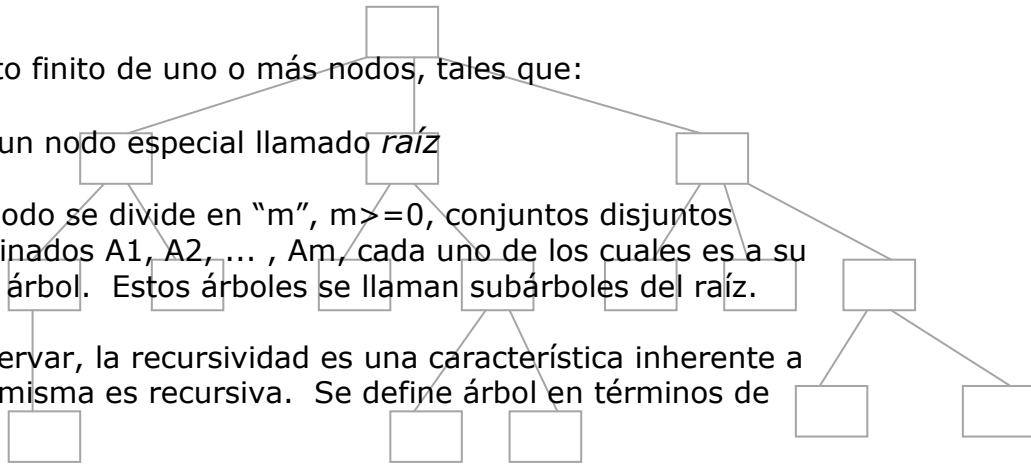
Es una estructura NO lineal, no fundamental, homogénea y finita.

## DEFINICIÓN

ÁRBOL: Es un conjunto finito de uno o más nodos, tales que:

- i) Existe un nodo especial llamado *raíz*
- ii) Cada nodo se divide en " $m$ ",  $m \geq 0$ , conjuntos disjuntos denominados  $A_1, A_2, \dots, A_m$ , cada uno de los cuales es a su vez un árbol. Estos árboles se llaman subárboles del raíz.

Como se puede observar, la recursividad es una característica inherente a los árboles. La definición misma es recursiva. Se define árbol en términos de árbol.



## TERMINOLOGÍA

NODO: A cada elemento de la estructura se le llama nodo. Cada nodo tiene UN ascendiente llamado "padre" y cero o más descendientes llamados "hijos". A los hijos de un mismo padre se les suele llamar "hermanos".

RAÍZ: Es el único nodo especial debido a que no tiene padre.

HOJA: Es un nodo que no tiene hijos.

INTERIOR: Todos los nodos que no son la raíz, ni son hojas, se llaman nodos interiores.

CAMINO: Es la secuencia de nodos por los que hay que pasar para llegar de un nodo a otro.

RAMA: Un camino que lleva hasta una hoja se llama rama.

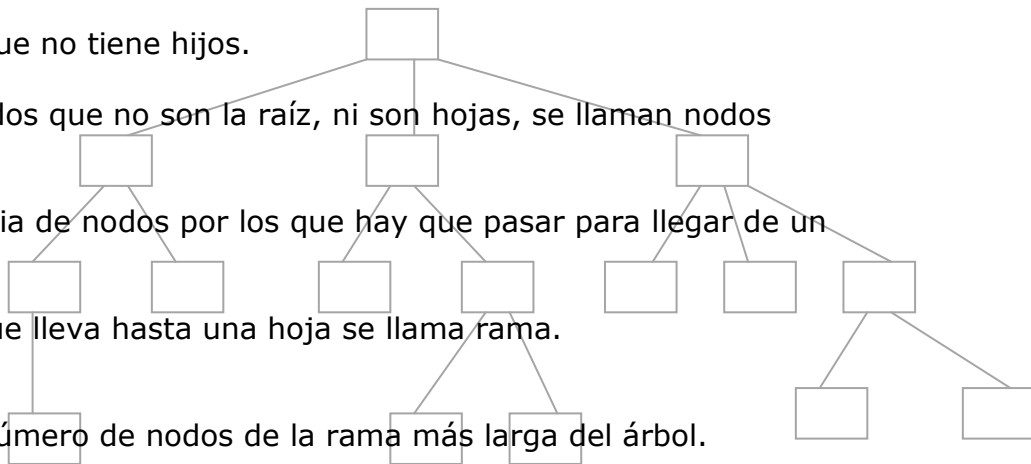
ALTURA O PROFUNDIDAD: Es el número de nodos de la rama más larga del árbol.

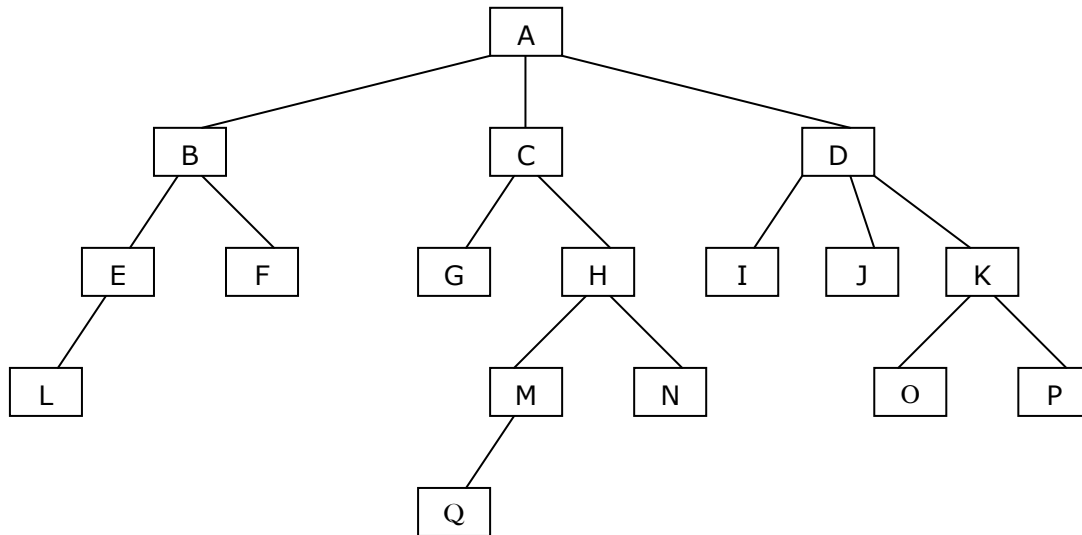
PESO: Es el número de hojas del árbol.

GRADO DE UN NODO: Es el número de hijos que tiene el nodo.

GRADO DEL ÁRBOL: Es el grado del nodo de mayor grado.

NIVEL: Cada nodo tiene asociado un nivel. El nivel de la raíz es 0. El nivel de cualquier nodo que no es raíz, es el nivel de su padre más 1.





RAIZ: **A.**

HOJAS: **L, F, G, Q, N, I, J, O, P.**

N. INTERIORES: **B, C, D, E, H, M, K.**

ALTURA: **5.**

PESO: **9.**

GRADO DEL NODO H: **2.**

GRADO DEL ÁRBOL: **3.**

NIVEL DE C: **1.**

NIVEL DE Q: **4.**

CAMINO DE C a Q: **C, H, M, Q** (Es rama porque lleva hasta una hoja).

RAMA MÁS LARGA: **A, C, H, M, Q .**

Todos los conceptos y definiciones que hemos expresado se refieren a “árboles generales”. Representar un árbol general mediante modelos computacionales reviste una complejidad importante a la vez que su utilidad no es la mayor.

La mayor utilidad se encuentra en los árboles binarios, mismos que son considerablemente más fáciles de implementar en la computadora.

## ÁRBOL BINARIO

Es un árbol en el que cada nodo puede tener 0, 1 o 2 hijos, conocidos como subárbol izquierdo y subárbol derecho. Es decir, un árbol de orden 2.

Existe alguna terminología propia de los árboles binarios.

## TERMINOLOGÍA

**BALANCEADO:** Se dice que un árbol binario está *balanceado* o *equilibrado*, cuando la diferencia de altura entre su subárbol izquierdo y su subárbol derecho es máximo 1. Puesto que cada nodo del árbol es a su vez un árbol, esta condición debe cumplirse para cada nodo.

**COMPLETO:** Un árbol binario es *completo* cuando cada uno de sus nodos tiene exactamente 2 hijos.

**SIMILAR:** Se dice que dos árboles binarios son *similares* si tienen la misma estructura.

**EQUIVALENTE:** Dos árboles binarios son *equivalentes* si son similares y tienen la misma información.

Vamos a estudiar las operaciones sobre los árboles binarios, por ser estos los que nos interesan más que los generales.

## OPERACIONES

Las operaciones que se pueden realizar con los árboles binarios son:

Recorrer: No se refiere a desplazamiento. Se refiere a visitar todos los nodos del árbol, sin repetir ninguno.

Insertar

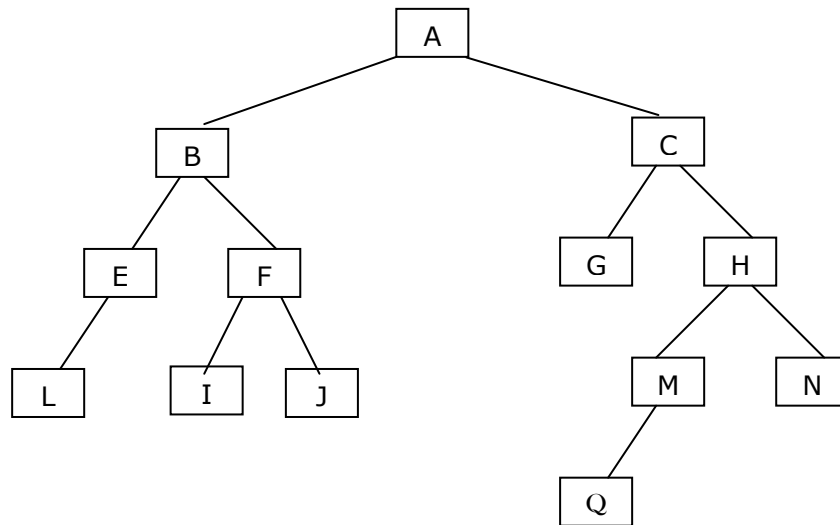
Borrar

Buscar

Ordenar

## RECORRER

Como podemos imaginar, existen múltiples formas para visitar todos los nodos de un árbol binario. De todas estas formas, se han adoptado 3, en las cuales siempre se recorre primero el subárbol izquierdo y después el derecho. En cada una de estas cambia el momento en que se visita la raíz, dando lugar a los recorridos: in-orden, pre-orden y post-orden. El prefijo de cada uno de ellos indica el turno de la raíz respecto a los dos subárboles.



## Recorrido **IN-ORDEN**

1. Recorrer subárbol Izquierdo en orden in-orden
2. Visitar Raíz
3. Recorrer subárbol Derecho en orden in-orden

Ej: .

## Recorrido **PRE-ORDEN**

1. Visitar Raíz
2. Recorrer subárbol Izquierdo en orden pre-orden
3. Recorrer subárbol Derecho en orden pre-orden

Ej: .

## Recorrido **POST-ORDEN**

1. Recorrer subárbol Izquierdo en orden post-orden
2. Recorrer subárbol Derecho en orden post-orden
3. Visitar Raíz

Ej: .

## ÁRBOLES DE EXPRESIÓN

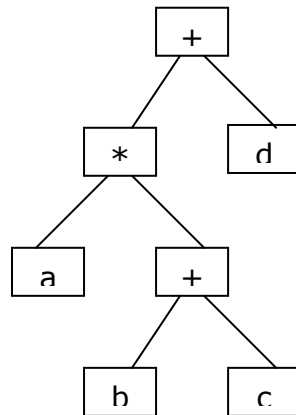
Los árboles de expresión son una aplicación de los árboles binarios. Son útiles para representar en memoria expresiones algebraicas en notación infija.

Un árbol de expresión es un árbol binario con las siguientes propiedades:

1. Cada hoja es un operando
2. Los nodos raíz y nodos interiores son operadores
3. Los subárboles son subexpresiones en las que el nodo raíz es un operador

Ejemplo:

$a * (b + c) + d$

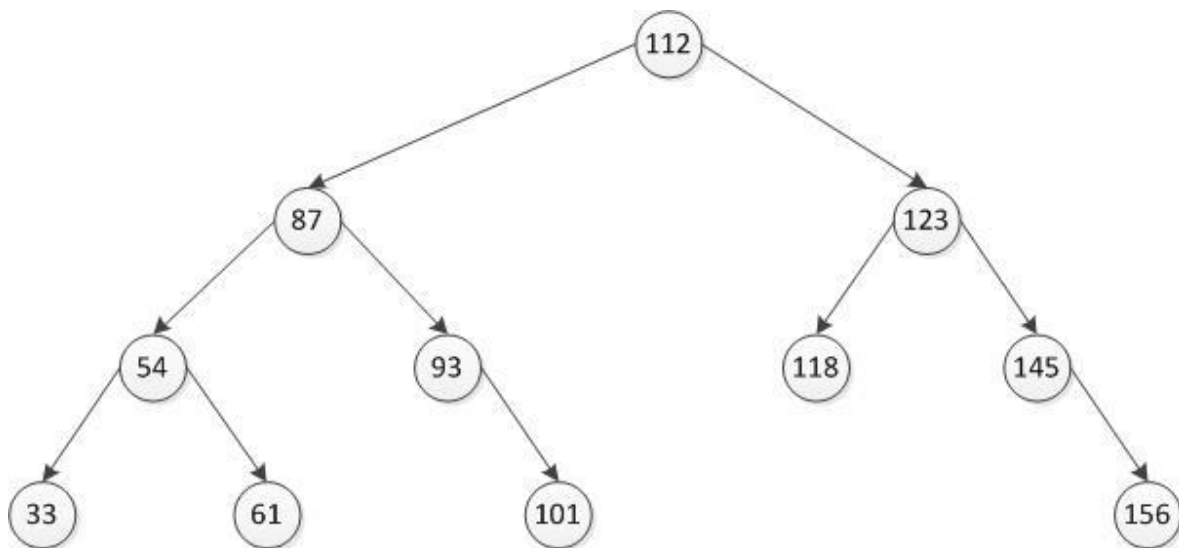




## ÁRBOL BINARIO DE BÚSQUEDA (ABB)

Es un árbol binario, en el que para cada nodo, los nodos de su subárbol izquierdo, son menores que él y los nodos de su subárbol derecho son mayores que él.

Ejemplo:



## ÁRBOL BINARIO DE BÚSQUEDA (ABB)

Una propiedad importante de los árboles binarios de búsqueda, es que al recorrerlos in-orden, la información se obtiene ordenada.

Ejemplo con el árbol de arriba:

33 54 61 87 93 101 112 118 123 145 156