Dra. Mayra CARRION.
mayra.carrion@epn.edu.ec

Árboles

Referencia: CAIRO, GUARDATI. "Estructura de datos"., ISBN: 970-10-0258-X. 2008
LUIS JOYANES AGUILAR. "Estructura de datos en JAVA". ISBN:9788448156312. 2008

Objetivos Sesión

- Compreder la definición Básica referente a Árboles
- Comprender las características principales de la estructura de datos tipo árbol.
- Comprender cada elemento que componen a esta estructura de datos.
- Analizar la representación de la estructura de datos tipo árbol.
- Analizar las diferentes lógicas algorítmicas que permiten la creación de la ED tipo ÁRBOL

¿Qué entienden por la estructura de datos tipo árbol ?

Árboles

Los árboles son las **estructuras de datos NO lineales y dinámicas** de datos más importantes del área de la computación .

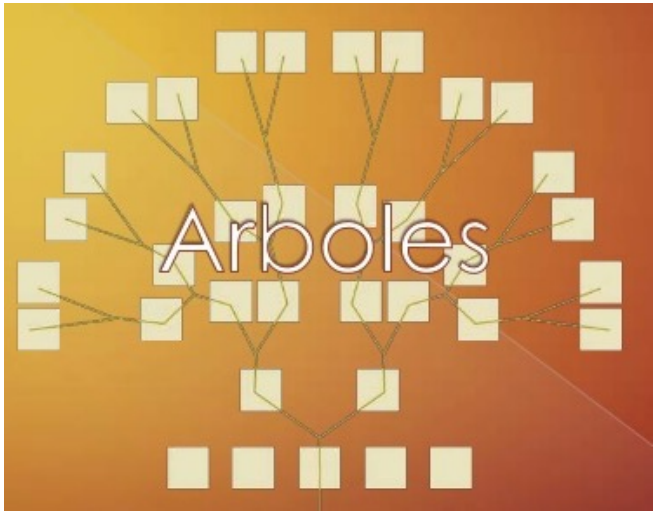
Los **árboles balanceados o AVL** son la estructura de datos más eficiente para trabajar con la **memoria principal** interna del procesador, mientras que los **árboles B y especialmente la versión B+**, representan la estructura de datos más eficiente para trabajar en **memoria secundaria o externa**.

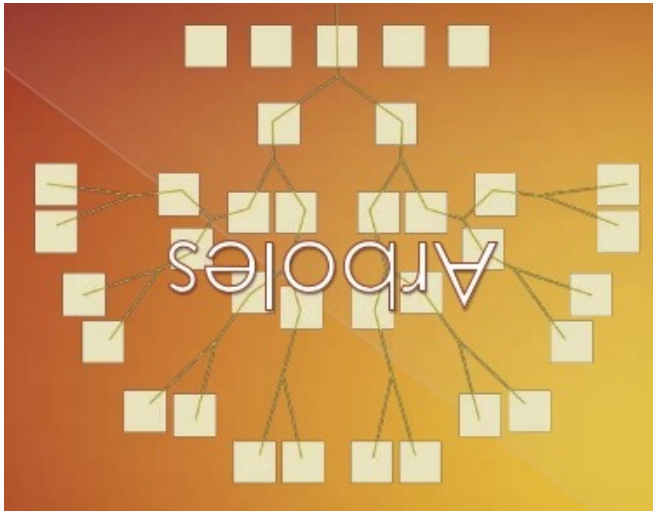
¿Cómo se la puede definir a la estructura de datos tipo árbol ?

ÁRBOLES

Un árbol se puede definir **como una estructura jerárquica** aplicada sobre una colección de elementos u objetos llamados nodos.

Un **árbol consta** de un **conjunto finito de elementos** , **denominados nodos**, y un conjunto **finito de líneas dirigidas denominadas ramas**, que conectan los nodos.

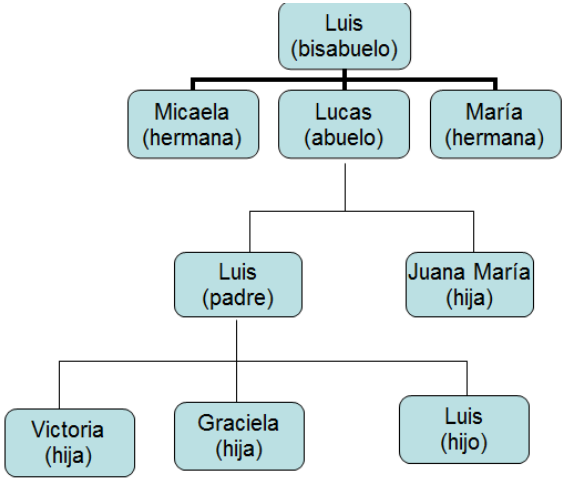


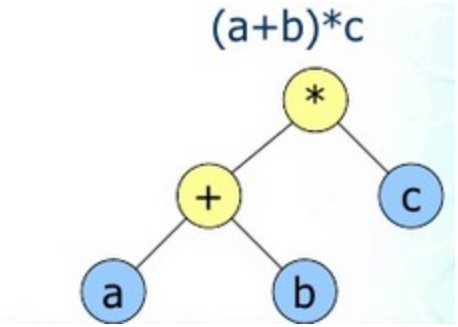


¿Ejemplos de aplicación de la estructura de datos tipo árbol ?

Ejemplo de Aplicación de Árboles

- Para representar **fórmulas matemáticas**
- Para registrar **historias de un campeonato** de tenis.
- Para construir un **árbol genealógico.**
- Para enumerar los **capítulos** y **secciones** de un libro.

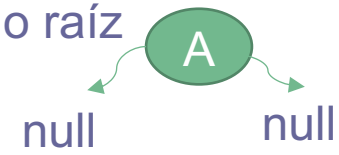




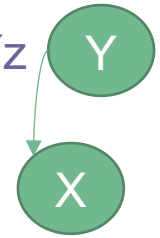
¿Características y Propiedades de la estructura de datos tipo árbol?

Características y Propiedades de los árboles

1. No no vacío Nodo raíz

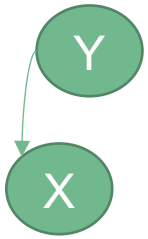


2. Nodo Raíz



Nodo X descendiente de nodo Y

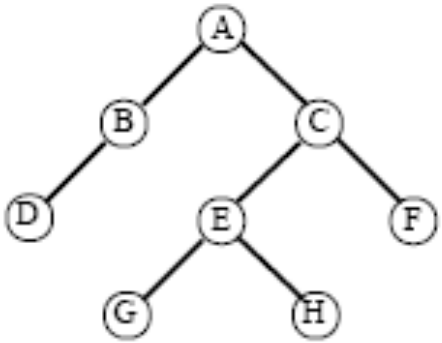
3. Cuando un nodo se considera nodo Padre. Y es nodo padre de Nodo X



4. Cuando los Nodos se consideran hermanos
5. Nodos Intreriores
6. Nodos Hojas o Nodos terminales
7. Grado del Nodo
8. Grado del Árbol
9. Nivel
10. Altura

Definiciones Básicas en Árboles

- *Nodo Padre*
- *Nodo Hijo*
- *Nodo Raíz*
- *Hojas*
- *Nodos Interiores*
- *Camino*
- *Rama*
- *Grado*
- *Niveles*



¿Características y Propiedades a visualizar en ejemplo?

¿Longitudes a considerar en la estructura de datos tipo árbol?

Longitud de camino interno y externo

De manera general las longitudes se define como el **número de arcos** que se **deben recorrer** para llegar **desde el nodo raíz hasta el nodo X**.

Por definición la raíz tiene longitud de **camino 1**, sus descendientes directos **longitud de camino 2**, etc.

Longitud de Camino Interno

- **LCI es la suma** de las longitudes de camino de todos los nodos del árbol.

- **Esta medida es importante porque nos permite conocer los caminos que tiene el árbol.**

- La fórmula que nos ayuda para calcular es la siguiente:

$$LCI = \sum_{i=1}^{h} n_i * i$$

Medida de Longitud de camino interno

Ahora para calcular LCIM . Se calcula dividiendo la LCI entre números de nodo del árbol (n) , esta medida es importante **porque nos ayuda a ver cuál es la mejor decisión** que debemos tomar para llegar a un nodo determinado partiendo del nodo raíz.

- Fórmula,

$$LCIM = LCI/n$$

Longitud de camino externo

Antes de definir la **LCE** es importante conocer lo que es **un árbol extendido y nodo especial**

Árbol extendido es aquel en el que el número de hijos de cada nodo es **igual al grado del árbol. Y si** alguno de los nodos no cumple con esta condición entonces deben incorporarse al mismo tantos nodos especiales como se requieran para llegar a cumplirla. **Los nodos especiales se** representan con cuadrados

$$LCE = \sum_{i=2}^{h+1} n_{ei} * i$$

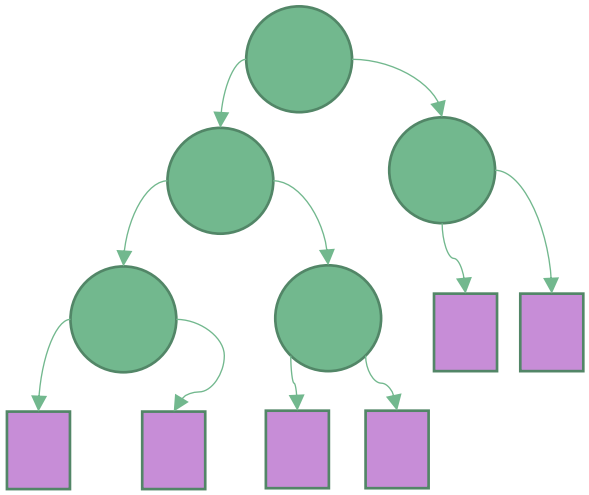
Medida de Longitud de camino extreno

LCEM se clacula a través de la siguiente fórmula:

$$LCEM=LCE/ne$$

Nos permite indicar el número de arcos que se deben recorrer en promedio para llegar, partiendo desde la raíz a un nodo especial cualquiera del árbol

Ejemplo de las Longitudes en la ED TIPO Árbol



Longitud Camino Interno
LCI= 1*1+2*2+2*3
LCI= 11
Medida LCI
LCIM= LCI/n= 11/5=2.5

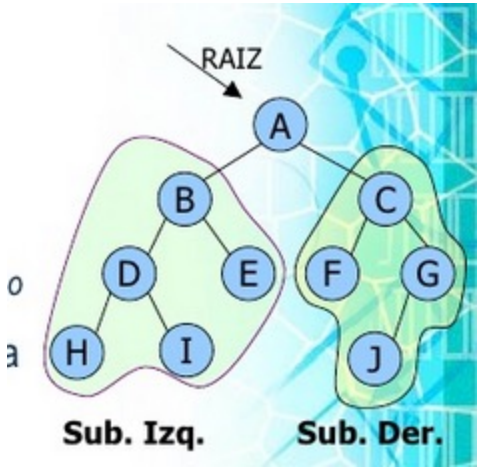
Longitud camino externo
LCE=2*3+4*4=22
LCE=22
Medida de LCE
LCEM=LCE/ne=22/6=3,6

ÁRBOLES BINARIOS?

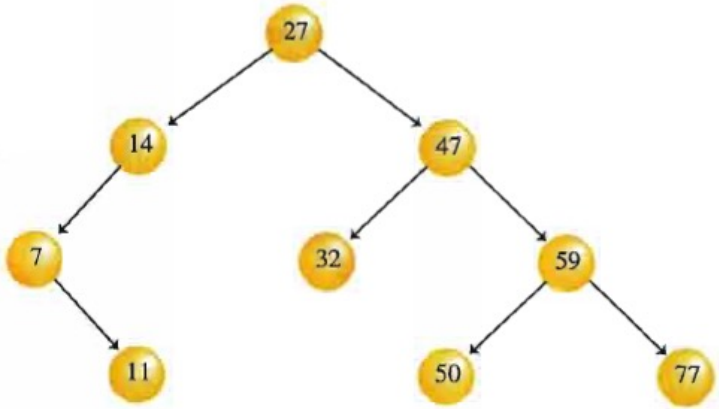
Árboles Binarios

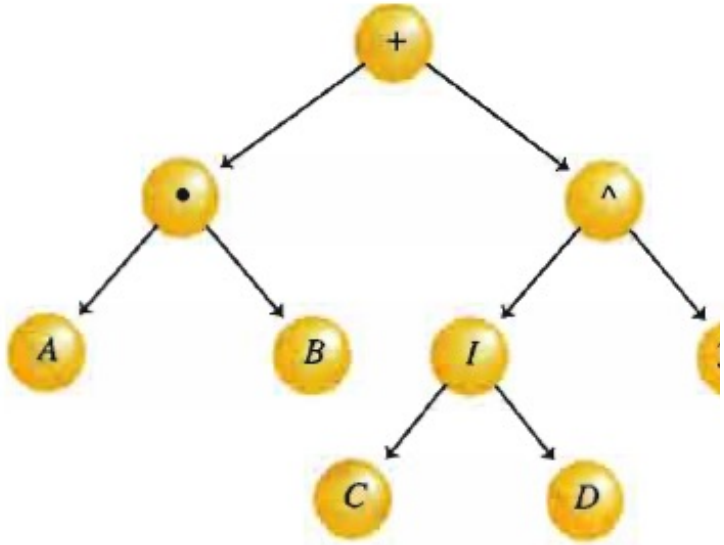
Este tipo de arboles son también conocidos como **árboles ordenados de grado 2.**

En un árbol binario cada nodo puede tener como **máximo dos subárboles** que se distinguen entre sí como el subárbol izquierdo y el subárbol derecho, según su ubicación con respecto al nodo raíz.



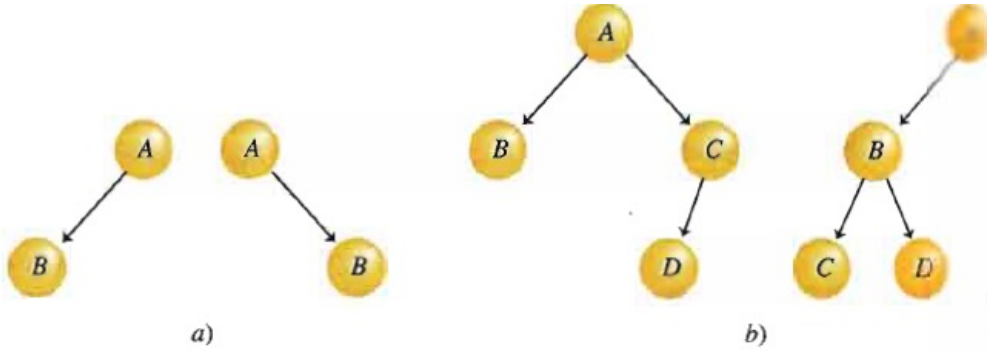
Ejemplos de Árboles Binarios



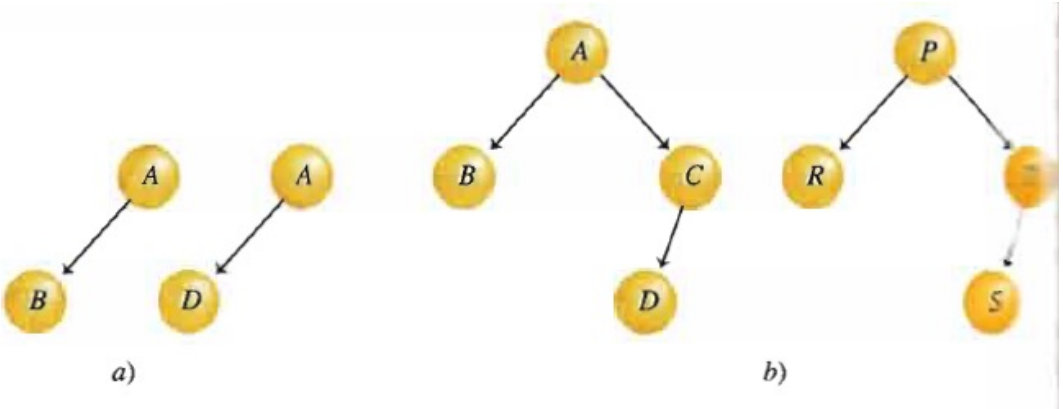


Árboles binarios distintos, similares y equivalentes

- **Árboles binarios distintos** : son diferentes sus estructuras-nodos- y arcos.

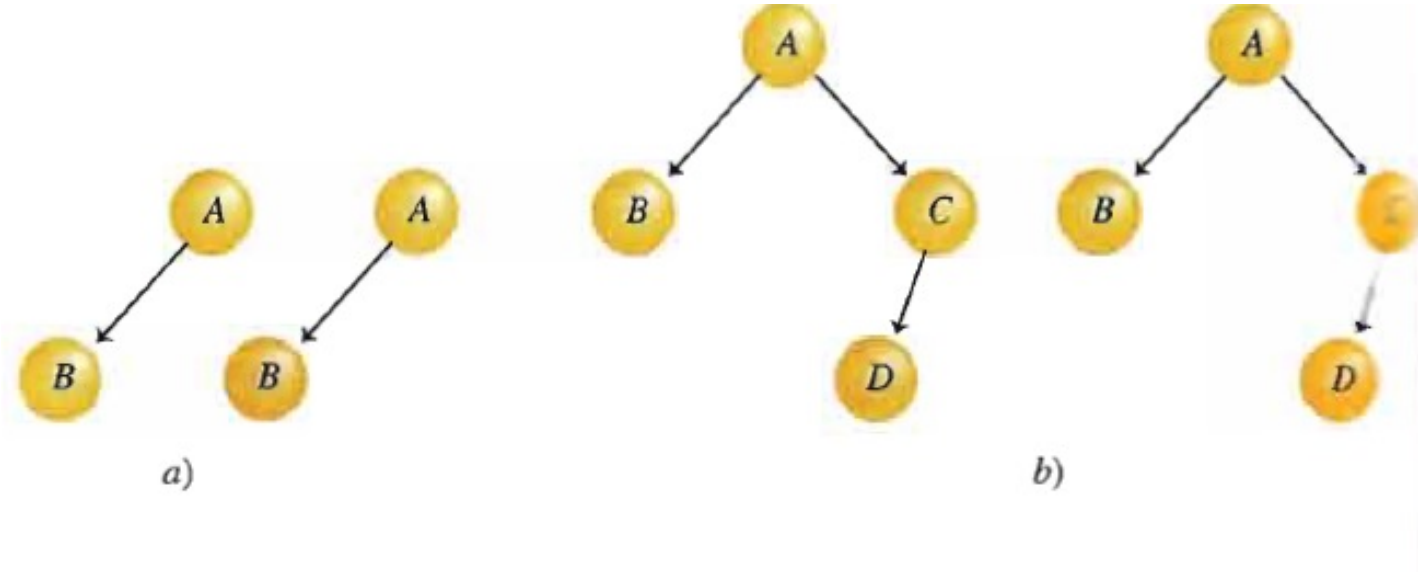


- **Árboles binarios similares** : las estructuras son idénticas solamente la información que contienen sus nodos difiere entre sí.



Árboles binarios distintos, similares y equivalentes

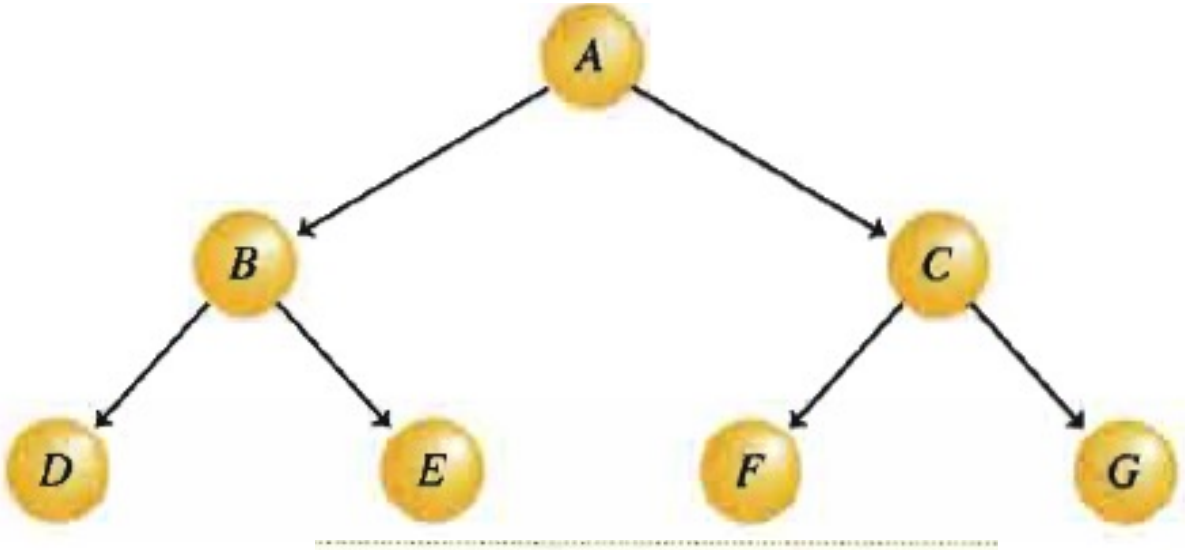
- **Árboles binarios equivalentes**: su estructura similar y además contienen la misma información.



Árboles binarios Completos

ABC se definen como un árbol en el que todos sus nodos **excepto los del último tienen dos hijos:** el subárbol izquierdo y el subárbol derecho.

Ejemplo de árboles binarios completos:



NÚMERO DE NODOS(ABC) = $2^h - 1$

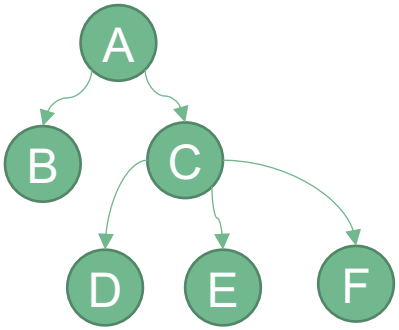
Representación de árboles generales como binarios

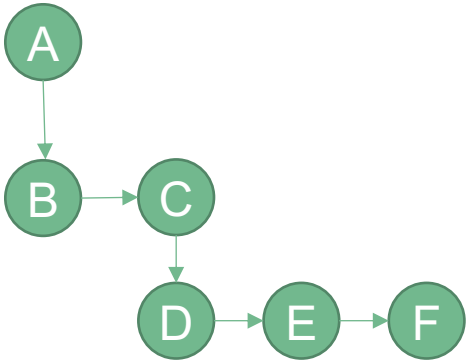
Es **importante el uso de árboles binarios** en el área de la computación es por ello que resulta de utilidad poder convertir **árboles generales** con o a n hijos en árboles binarios.

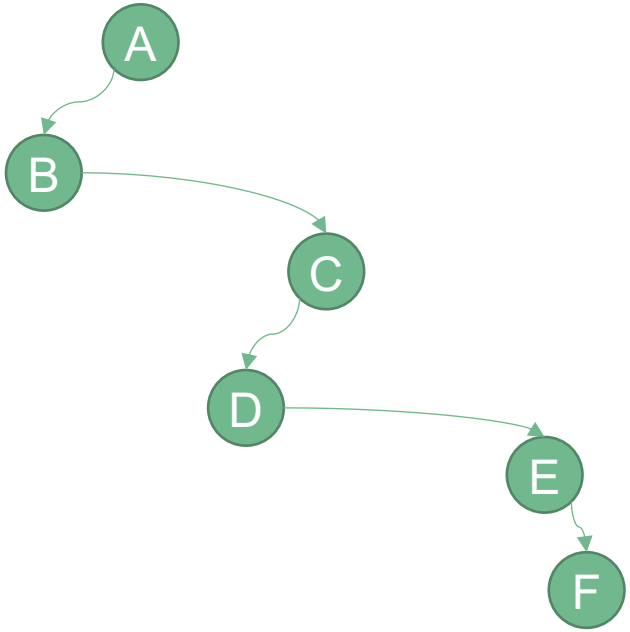
Pasos a considerar:

1. Enlazar los hijos de cada nodo en forma horizontal —los hermanos—.
2. Relacionar en forma vertical el nodo padre con el hijo que se encuentra más a la izquierda. Además, se debe eliminar el vínculo de ese padre con el resto de sus hijos.
3. Rotar el diagrama resultante, aproximadamente 45 grados hacia la izquierda, y así se obtendrá el árbol binario correspondiente.

Ejemplo de Árboles Generales a Árboles Binarios





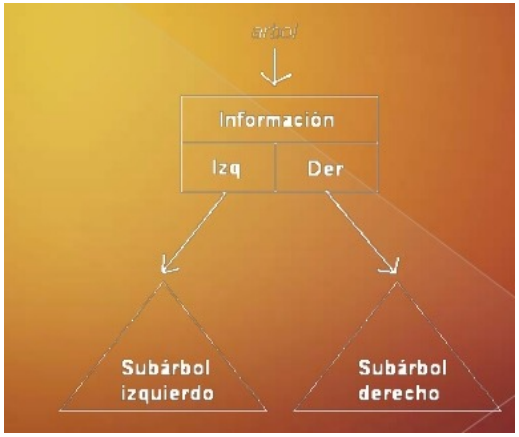


Representación de árboles binarios en memoria

Existen dos manera más comunes de representar un árbol binario en memoria que son las siguientes:

- Por medio de arreglos
- **Por medio de datos tipo puntero,** también conocidos como **variables dinámicas**

Dado el nodo *T*:



Descripción de campos del NODO

IZQ: Campo donde se almacena la dirección del subárbol izquierdo del nodo T.

INFO: Representa el campo donde se almacena la información del nodo.

DER: Campo que almacena la dirección del subárbol derecho del nodo.

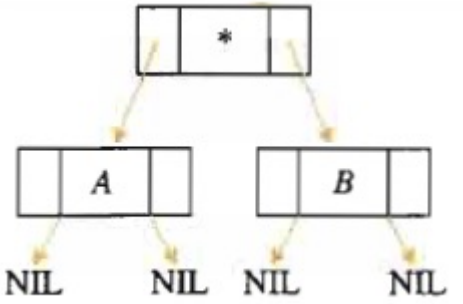
Definición de árbol binario

- Definición de un árbol binario en Lenguaje Algorítmico:

```
ENLACE=^.NODO
NODO= REGISTRO
        IZQ: TIPO ENLACE
        INFO: TIPO DATO
        DER: TIPO ENLACE
FIN DEFINICION
```

Ejemplo de representación en memoria

Consideremos el árbol binario que permita representar la siguiente expresión algebraica A*B. Su representación de la estructura de datos cómo sería en memoria??:



Obsérvese que pasa con el término NIL se utilizará o no ?

¿Cómo se crearía una estructura de datos tipo árbol. Con la ayuda de un algoritmo?

Creación de Arboles Binarios

Crear_arbol (APNODO)

{El algoritmo crea un árbol binario en memoria. APNODO es una variable de tipo ENLACE —puntero a un nodo—. La primera vez APNODO se crea en el programa principal}
{INFO, IZQ y DER son campos del registro NODO. INFO es de tipo carácter. IZQ y DER son de tipo puntero. Las variables RESP y OTRO son de tipo carácter y de tipo ENLACE, respectivamente}

1. Leer APNODO^.INFO {Lee la información y se guarda en el nodo}
2. Escribir "¿Existe nodo por izquierda: 1(Sí) – 0(No)?"
3. Leer RESP
4. *Si* (RESP = "Sí")
   entonces
      Crear(OTRO) {Se crea un nuevo nodo}
      Hacer APNODO^.IZQ ← OTRO
      Regresar a Crea_árbol con APNODO^.IZQ {Llamada recursiva}
   si no
      Hacer APNODO^.IZQ ← NIL
5. {Fin del condicional del paso 4}
6. Escribir "¿Existe nodo por derecha: 1(Sí) – 0(No)?"
7. Leer RESP
8. *Si* (RESP = "Sí")
   entonces
      Crear(OTRO) {Se crea un nuevo nodo}
      Hacer APNODO^.DER ← OTRO
      Regresar a Crea_árbol con APNODO^.DER {Llamada recursiva}
   si no
      Hacer APNODO^.DER ← NIL
9. {Fin del condicional del paso 8}

¿Recorrido en árboles binarios?

Recorrido en árboles

Una de las operaciones más importantes que se realiza en un árbol binario es el recorrido de los mismos .

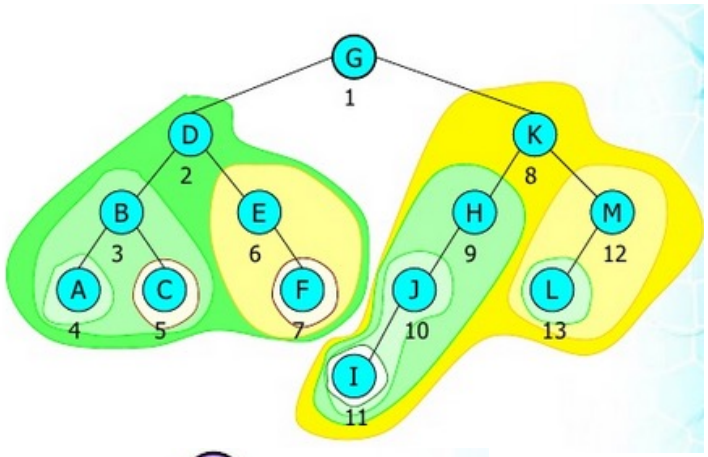
Tres formas diferentes de efectuar el recorrido :

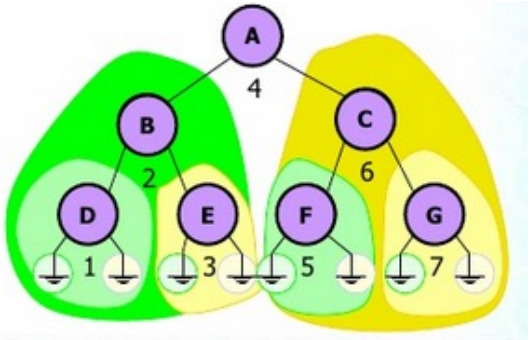
RECORRIDO EN PREORDEN

- Visitar la raíz
- Recorrer el subárbol izq
- Recorrer el subárbol der

RECORRIDO EN INORDEN

- Recorrer el subárbol izq
- Visitar la raíz
- Recorrer el subárbol der





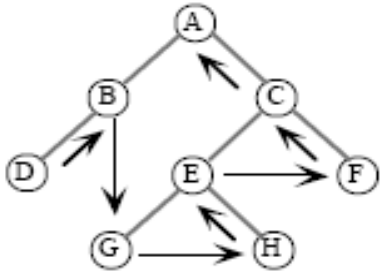
Algoritmos de Recorrido en ABB

- **RECORRIDO EN POSORDEN**

Recorrer el subárbol izq

Recorrer el subárbol der

Visitar la raíz



Algoritmos de Recorrido PreOrden

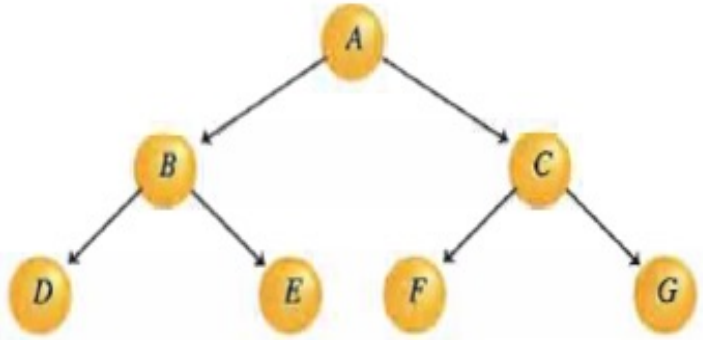
PREORDEN (APNODO)

PreOrden(APNODO)
INFO
IZQ
DER
Si (APNODO!= Nul)
Visita el APNODO Escribir APNODO^.INFO

Regresar a PreOrden (APNODO^.IZQ)

Regresar PreOrden (APNODO^.DER)

Fin SI
Fin PreOrden



Algoritmos de Recorrido InOrden

INORDEN (APNODO)

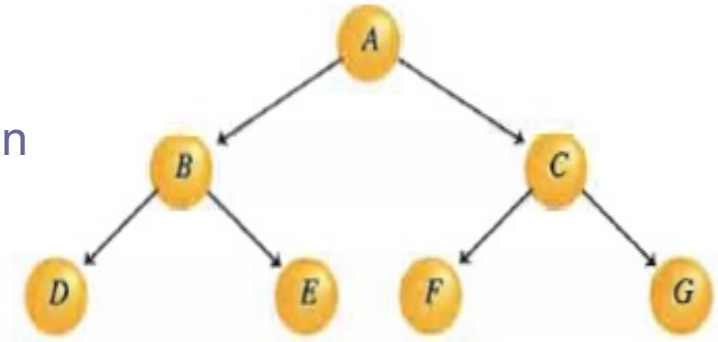
InOrden(APNODO)
INFO
IZQ
DER
Si (APNODO!= Nul)

Regresar a InOrden (APNODO^.IZQ)

Visita el APNODO Escribir APNODO^.INFO

Regresar a InOrden (APNODO^.DER)

Fin SI
Fin InOrden



Algoritmo de Recorrido en PostOrden

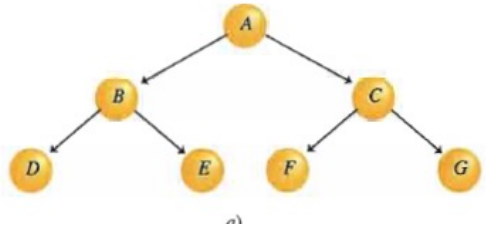
POSORDEN (APNODO)

PosOrden(APNODO)
INFO
IZQ
DER
Si (APNODO!= Nul)

Regresar a PosOrden (APNODO^.IZQ)

Regresar a PosOrden (APNODO^.DER)

Visita el APNODO Escribir APNODO^.INFO
Fin SI
Fin InOrden



Gracias

¿…?