



VERACRUZ  
GOBIERNO DEL ESTADO



SEV  
Secretaría  
de Educación



VERA  
CRUZ  
ME LLENA DE ORGULLO



DET  
Dirección de Educación  
Tecnológica del Estado  
de Veracruz



# Instituto Tecnológico Superior de Chicontepec

Ingeniería en Sistemas Computacionales

**Nombre:**

Camelia Bautista Hernández

**Docente:**

Ing. Efrén Flores Cruz

**Asignatura:**

Programación Lógica y Funcional

**Unidad 2**

Modelo de programación funcional

**Actividad 1 Resumen**

**9.1 Arboles Generales**

**9.2 Arboles Binarios**

**9.3 Arboles de Búsqueda**

**8º SEMESTRE**

## ACTIVIDAD 1 ARBOLES

### 9.1 Arboles Generales

Tema: Arboles

DIA	MES	AÑO	FOLIO
31	03	20	

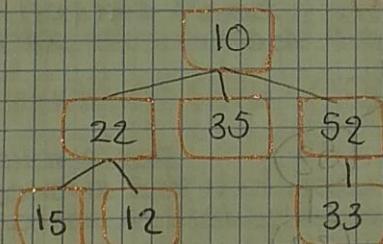
#### 9.1 Arboles Generales

Un arbol es una estructura no lineal jerárquica utilizada para organizar información de forma eficiente.

La definición es recursiva:

Un arbol es una colección de valores  $\{v_1, v_2, \dots, v_n\}$  tales que si  $n = 0$  el arbol se dice vacío.

En otro caso existe un valor destacado que se denomina raíz (p. e.  $v_1$ ), y los demás elementos forman parte de colecciones disjuntas que a su vez son árboles. Estos árboles se llaman subárboles de la raíz.



Las estructuras tipo arbol se usan principalmente para representar datos con una relación jerárquica entre sus elementos, como árboles genealogicos, tablas etc.

La terminología de los árboles se realiza con las típicas notaciones de las relaciones familiares en los árboles genealogicos: padre, hijo, hermano, ascendente, descendiente, etc.

Nodo son elementos de arbol

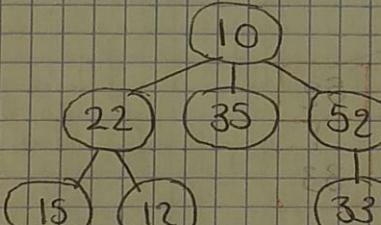
Raiz del Arbol: todos los árboles que no están vacíos tienen una única nodo raiz. Todos los demás elementos o nodos se derivan o descienden de el.

DIA / MES / AÑO / FOLIO

Los 80 18

- ⑥ Nodo hoja es aquel nodo que no tiene ningún <sup>sub</sup>árbol.
- ⑥ Tamaño de árbol es un número de nodos.
- ⑥ A cada nodo que no es hoja se le asocia uno o varios subárboles llamados descendientes o hijos.
- ⑥ De igual forma, cada nodo tiene asociado un antecesor o ascendiente llamado padre.
- ⑥ Todos los nodos tienen un solo padre excepto el raíz que no tiene padre.
- ⑥ Cada nodo tiene asociado un número de nivel que se determina por la longitud del camino desde el raíz al nodo específico.

**Ejemplo**



```

graph TD
    10 --- 22
    10 --- 35
    22 --- 15
    22 --- 12
    35 --- 33
  
```

Raíz : 10

Nodos : 10, 22, 35, 52, 15, 12, 33

Tamaños : 7

Nivel : 10

Nivel 1: 22, 35, 52

Nivel 2: 15, 12, 33

Altura o profundidad : 3

Hojas: 15, 12, 35, 33

## 9.2 Arboles Binarios

9.2- Arboles Binarios

DIA	MES	AÑO	FOLIO
31	03	20	clase

Un arbol binario es árbol tal que cada nodo tiene como máximo dos subárboles.

### Recorrido de árboles binarios

Se llama recorrido de un árbol al proceso que permite acceder una sola vez a cada uno de los nodos del árbol para examinar el conjunto completo de nodos.

\* Los algoritmos de recorrido de un árbol binario presentan 3 tipos de actividades comunes

- ➊ Visitar el nodo raíz
- ➋ Recorrer el subárbol izquierdo
- ➌ Recorrer el subárbol derecho

\* Estas 3 acciones llevadas a cabo en distinto orden proporcionan los distintos recorridos de Árbol.

### Recorrido en PRE-ORDEN

- ➊ Visitar el raíz
- ➋ Recorrer el subárbol izquierdo en pre-orden
- ➌ Recorrer el subárbol derecho en pre-orden

### Recorrido EN-ORDEN

- ➊ Recorrer el subárbol izquierdo en en-orden
- ➋ Visitar el raíz
- ➌ Recorrer el subárbol derecho en en-orden

### Recorrido en POST-ORDEN

- ➊ Recorrer el subárbol izquierdo en post-orden
- ➋ Recorrer el subárbol derecho en post-orden
- ➌ Visitar el raíz



TECNM  
TECNOLÓGICO NACIONAL DE  
MÉXICO



VERACRUZ  
GOBIERNO DEL ESTADO



SEV  
Secretaría  
de Educación



VERA  
CRUZ  
ME LLENA DE ORGULLO



DET  
Dirección de Educación  
Tecnológica del Estado  
de Veracruz



DIA MES AÑO FOLIO

Ejemplo

```
graph TD; 10((10)) --> 15((15)); 10 --> 18((18)); 15 --> 24L((24)); 15 --> 27((27)); 18 --> 24R((24))
```

data ÁrbolB a = VacíoB | NodoB (ÁrbolB a) a (ÁrbolB a)

denuing Show

Consideraremos que las tres componentes del constructor NodoB son el subárbol izquierdo, el dato raíz y el subárbol derecho respectivamente.

```
graph TD; 10((10)) --> 15((15)); 10 --> 18((18)); 15 --> 27((27)); 15 --> 24L((24)); 18 --> 24R((24))
```

a2 :: Árbol Integer

a2 = NodoB a1 10 a3

where

a1 = NodoB a11 15 a10

a0 = NodoB VacíoB 18 a00

a11 = hojaB 24

a10 = hojaB 27

a00 = hojaB 24

hojaB :: a → ÁrbolB

hojaB x = NodoB VacíoB x VacíoB

## 9.3 Árboles de Búsqueda

### 9.3 Árboles de Búsqueda

DIA      MES      AÑO      FOLIO  
 31      03      20

Un árbol de búsqueda es un árbol binario tal que:

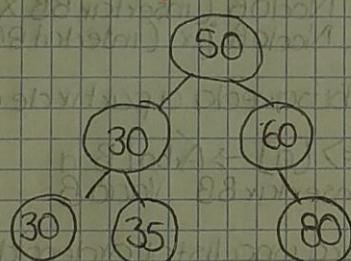
O bien es vacío.

O bien es vacío y para cualquier nodo se cumple que:

los elementos del subárbol izquierdo son menores o iguales al almacenado en el nodo.

Y los elementos del subárbol derecho son estrictamente mayores al almacenado en el nodo.

#### Ejemplo



La siguiente función puede ser utilizada para comprobar si un árbol binario es de búsqueda.

es ÁrbolBB

$\bullet\bullet$  Ord q  $\Rightarrow$  ÁrbolB al  $\rightarrow$  Bool

es ÁrbolBB VacíoB

= True

es ÁrbolBB (NodoB i v d)

= todos ÁrbolB ( $<=$  r)

el todos ÁrbolB ( $<$  r) d

el es ÁrbolBB r

el es ÁrbolBB d

todos ÁrbolB

$\bullet\bullet$  ( $a \rightarrow$  Bool)  $\rightarrow$  ÁrbolB d  $\rightarrow$  Bool

todos ÁrbolB p VacíoB

= True

todo ÁrbolB p (NodoB i rd)

= p x el

todos ÁrbolB p i el todos ÁrbolB pd

DIA      MES      AÑO      FOLIO

OS 80 18

④ Pertenencia a un árbol de búsqueda

Pertenece BB  
 Pertenece BB x Vacío B  
 Pertenece BB x (Nodo B i v d)  
 | x = v  
 | x < y  
 | otherwise

∴ Ord a  $\Rightarrow$  a  $\rightarrow$  ArbolBa  $\rightarrow$  Bool  
 = False  
 = True  
 pertenece BB x i  
 pertenece BB x d

⑤ Inserción en un árbol de búsqueda

Insertar BB  
 Insertar BB x Vacío B  
 Insertar BB x (NodoB i v d)  
 | x <= v  
 | otherwise

∴ Ord a  $\Rightarrow$  a  $\rightarrow$  ArbolBa  $\rightarrow$  Arbol Bq  
 = NodoB VacíoB x VacíoB  
 = NodoB (Insertar BB x i) x d  
 NodoB (x (Insertar BB x d))

⑥ Construcción de un árbol de búsqueda a partir de una lista

lista AA x b d BB    ∴ Ord a  $\Rightarrow$  [a]  $\rightarrow$  Árbol B d  
 listaAA x b o l BB = foldr insertar BB Vacío B

⑦ El recorrido en orden genera una lista ordenada (tree sort)

treeSort    ∴ Ord a  $\Rightarrow$  [a]  $\rightarrow$  [a]  
 treeSort = enOrdenB . lista AA x b o l BB

? treeSort [4, 7, 1, 2, 9]  
 [1, 2, 4, 7, 9] ∵ [Integer]