



Árboles Binarios Ordenados

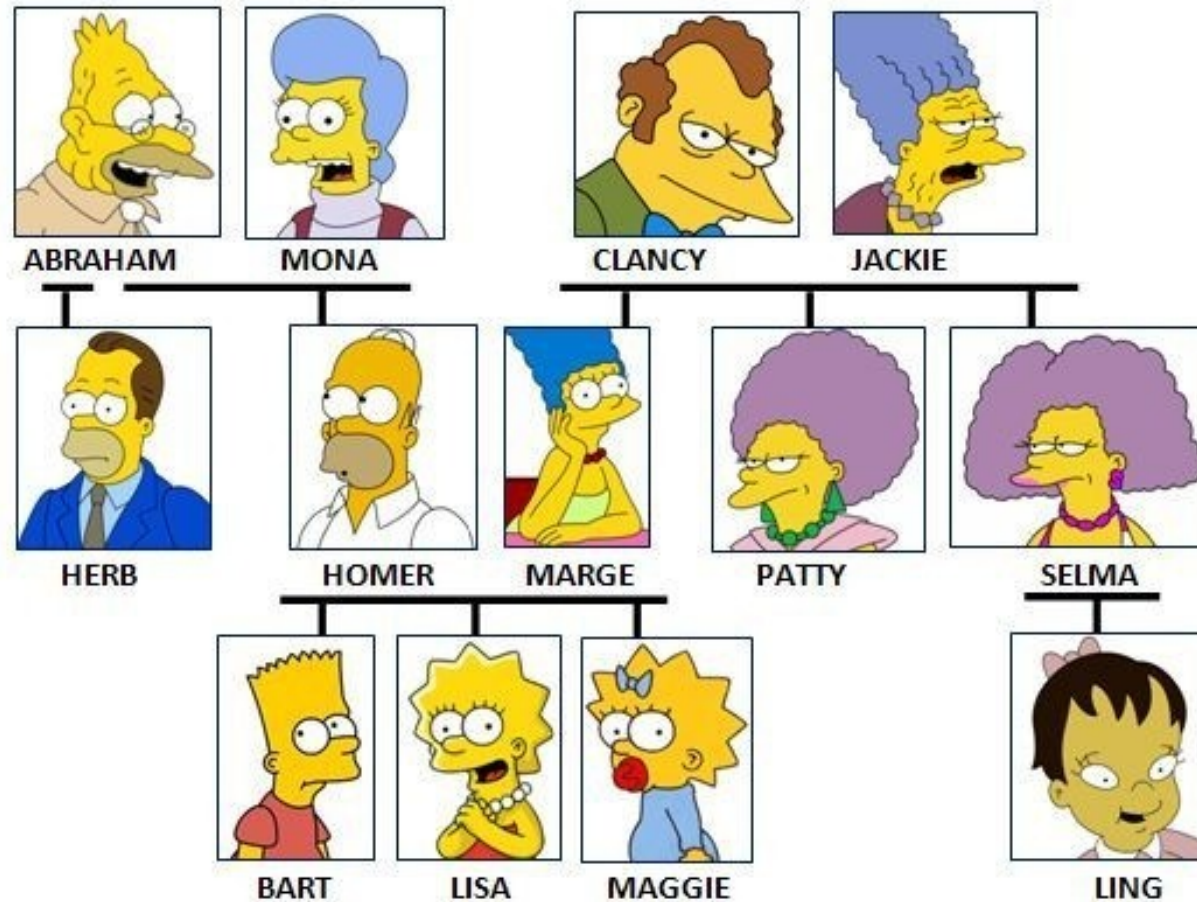
Temas de la clase

1. Árboles. Definición y características.
2. Operaciones con árboles.

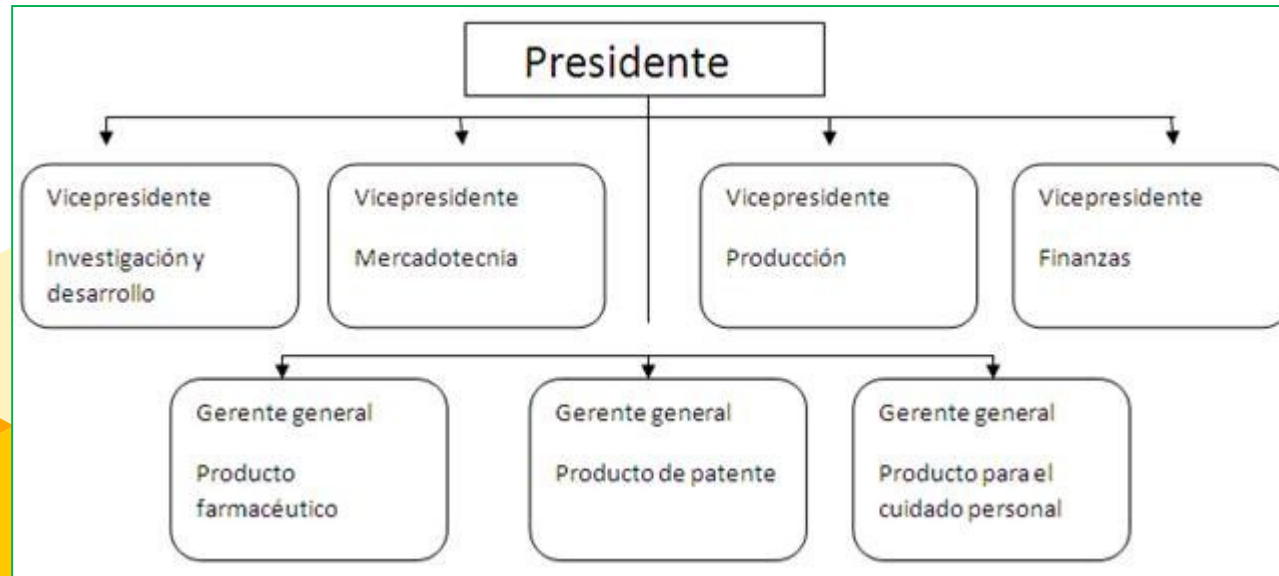
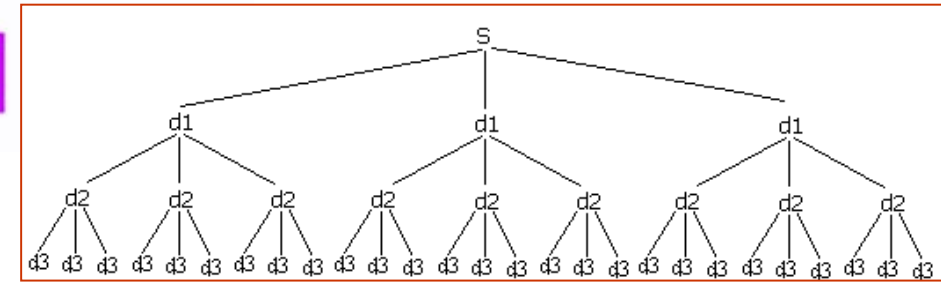
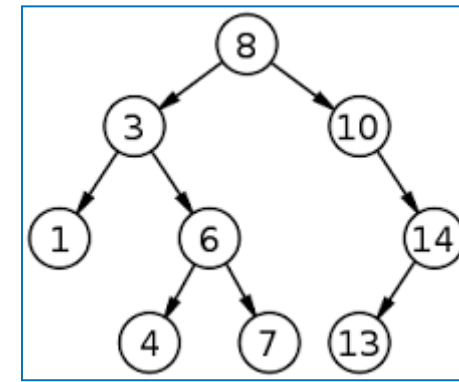
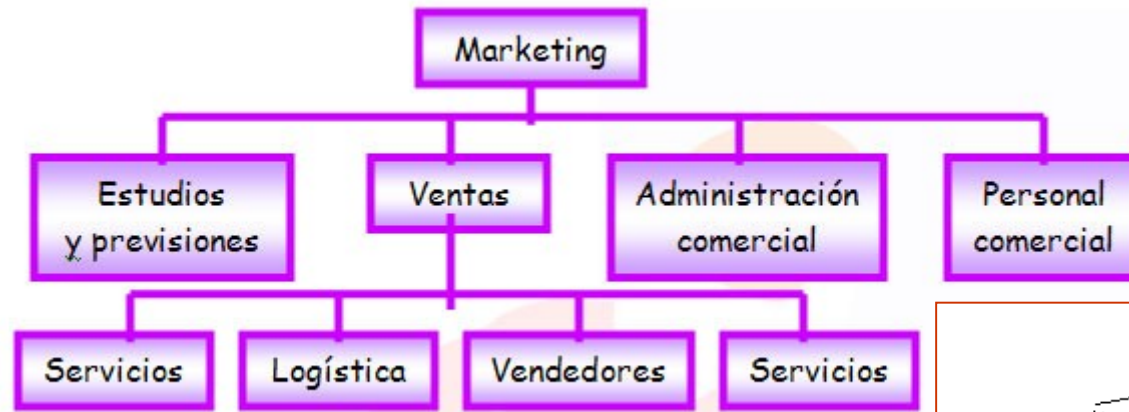


Árboles en general

LOS SIMPSONS



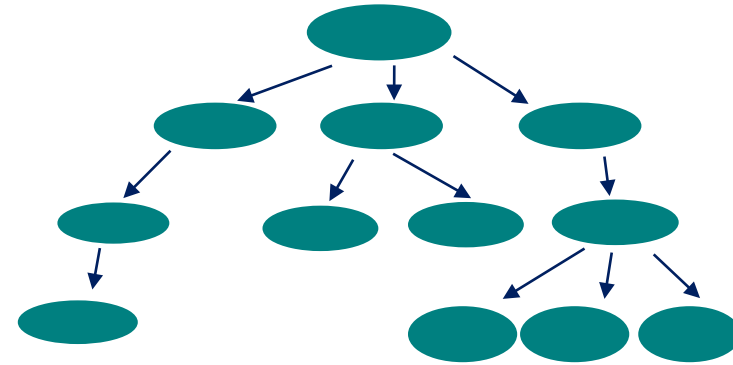
Árboles en general



ÁRBOLES en computación

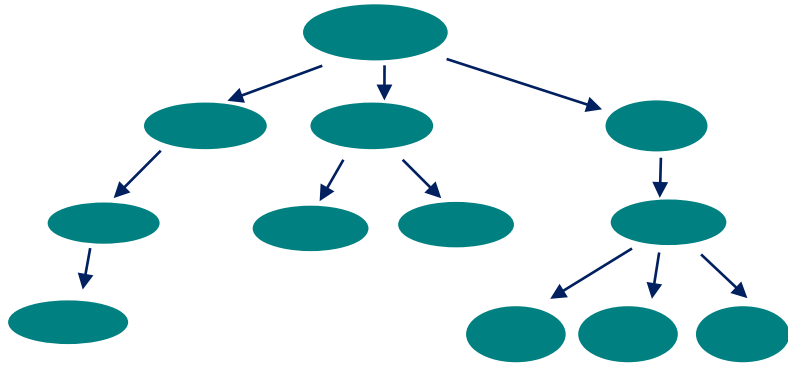
Un árbol es una estructura de datos que satisface tres propiedades:

- Cada elemento del árbol se relaciona con cero o más elementos (hijos).



- Si el árbol no está vacío, hay un único elemento (raíz) y que no tiene padre (predecesor).
- Todo otro elemento del árbol posee un único padre y es un descendiente de la raíz.

ÁRBOLES en computación - Características



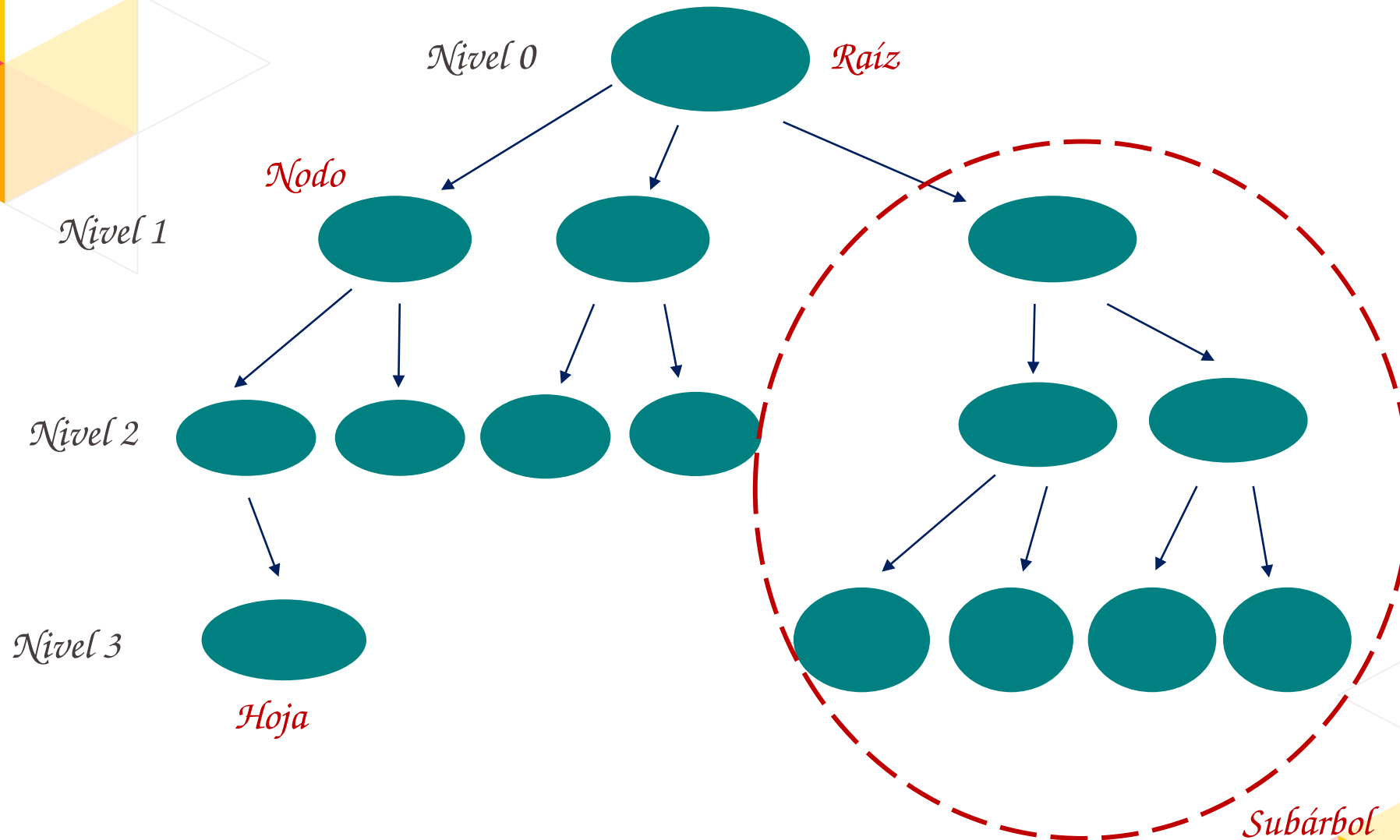
1. **Homogénea:** todos los elementos son del mismo tipo

2. **Dinámica:** puede aumentar o disminuir su tamaño durante la ejecución del programa

3. **No lineal:** cada elemento puede tener 0, 1, o más sucesores

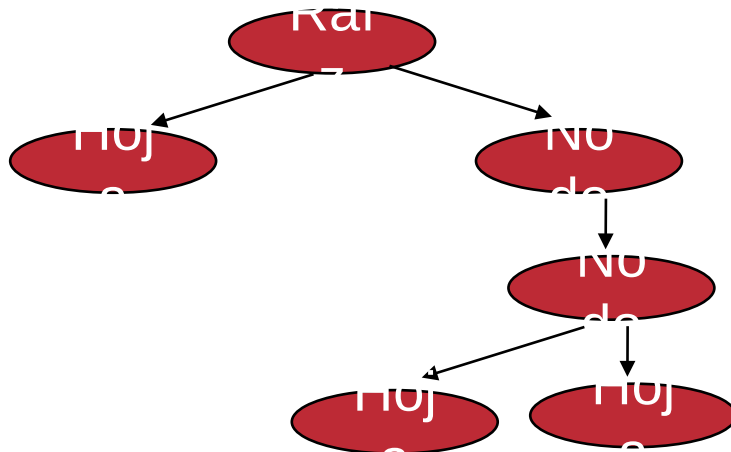
4. **Acceso Secuencial**

ÁRBOLES en computación - Terminología

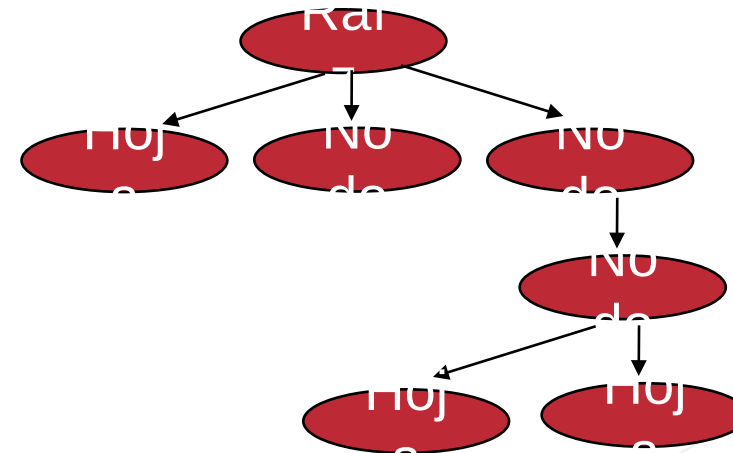


ÁRBOLES - Conceptos

Cuando cada nodo tiene como máximo 2 hijos se denominan árboles **BINARIOS**.



Cuando cada nodo tiene como máximo 3 hijos se denominan árboles **TERNARIOS**.



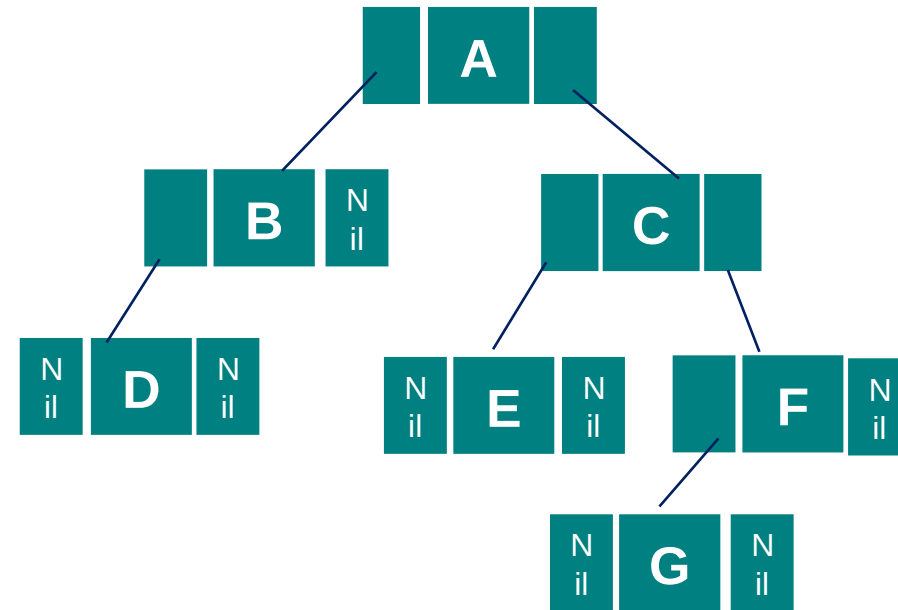
Cuando cada nodo tiene n hijos se llaman árboles **N-ARIOS**

ÁRBOLES BINARIOS

¿Cómo se relacionan los nodos de un árbol binario?

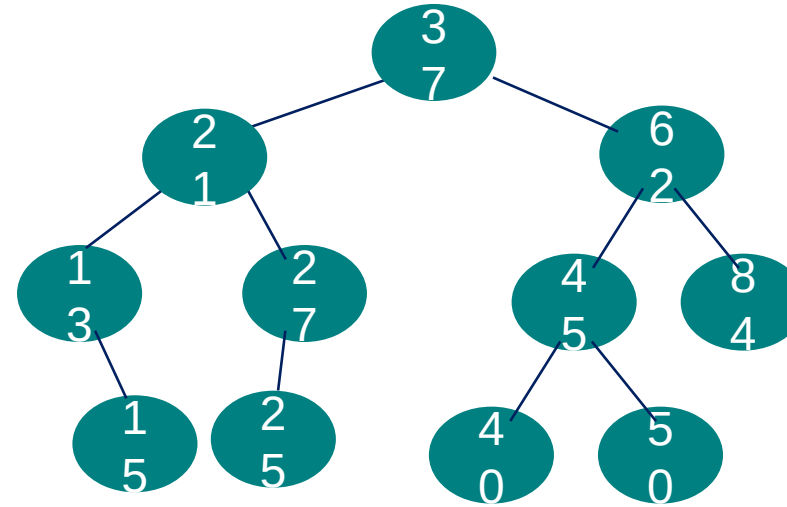
Type

```
elemento = tipoElemento;  
arbol = ^nodo;  
  
nodo = record  
  elem: elemento;  
  hijoIzq: arbol;  
  hijoDer: arbol;  
end;
```



Árboles Binarios Ordenados(ABO)

Observaciones



1. Cada nodo tiene un valor que:

- Es más grande que el valor de todos los nodos del subárbol izquierdo
- Es menor que el valor de todos los nodos del subárbol derecho

2. Utilidad más importante → Búsquedas el tiempo medio es $O(\log n)$



¿Cómo funciona el agregado de un nodo a un árbol binario ordenado?

http://163.10.22.82/OAS/AVL_Insercion/aplicacin_para_construir_un_rbol_avl.html

ABO – Operación Insertar Nodo

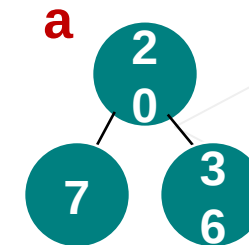
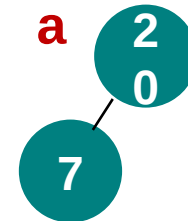
Supongamos que se leen los números: 20 7 36 1 4 23

2
0 Inicialmente **a** es nil.
El nuevo nodo se convierte en la raíz del árbol. HI y HD en Nil.

7 **a** no es Nil. Debe ubicarse donde insertar.
Como $7 < 20$ se toma el subárbol izquierdo de 20. Como el HI de 20 es Nil, se inserta.

3
6 **a** no es Nil. Debe buscar el lugar donde insertar.
Como $36 > 20$ se toma el subárbol derecho. Como el HD de 20 es Nil se inserta.

Siempre
se agrega
a nivel de
hoja
a



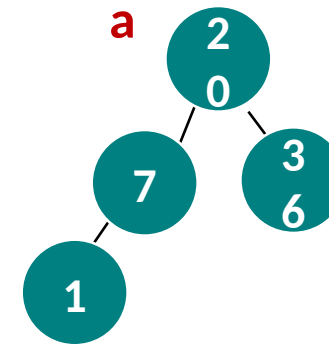
ABO – Operación Insertar Nodo

Supongamos que se leen los números: 20 7 36 1 4 23

1

a no es Nil.

Como $1 < 20$ se toma el subárbol izquierdo. Como el HI no es Nil, se vuelve a comparar y como $1 < 7$. Se elige el subárbol izquierdo y como es Nil se inserta.



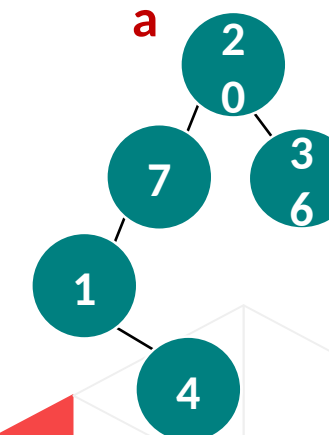
4

a no es Nil.

Como $4 < 20$ se toma el subárbol izquierdo.

Como el HI no es Nil y $4 < 7$ se elige el subárbol izquierdo.

Luego, como $4 > 1$, se elige el subárbol derecho. Como el nil se inserta.



ABO – Operación Insertar Nodo

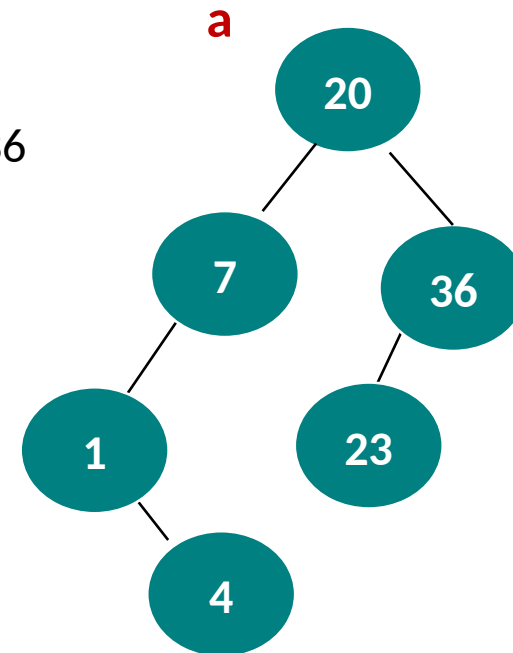
Supongamos que se leen los números: 20 7 36 1 4 23

a no es Nil.

Como $23 > 20$ se toma el subárbol derecho.

Como el HI no es Nil se vuelve a comparar y como $23 < 36$ se elige el subárbol izquierdo. Como es Nil se inserta

2
3





Insertar un dato

```
insertar (arbol, dato)
  si arbol es nil
    creo nodo_nuevo y pongo el dato y los hijos en
    nil
    arbol := nodo_nuevo
  sino
    si el dato en árbol es > dato
      insertar(hijo_izquierdo_del_árbol, dato);
    sino
      insertar(hijo_derecho_del_árbol, dato);
```

¿Qué pasa si los valores a insertar estuvieran repetidos?



Insertar un dato

```
insertar (arbol, dato)
  si arbol es nil
    creo nodo_nuevo y pongo el dato y los hijos en
    nil
    arbol := nodo_nuevo
  sino
    si el dato en árbol es > dato
      insertar(hijo_izquierdo_del_árbol, dato);
    sino
      si el dato en árbol es < dato
        insertar(hijo_derecho_del_árbol, dato);
```





Actividades en Máquina

- Descargar de Asignaturas: **ProgramaGenerarLista**

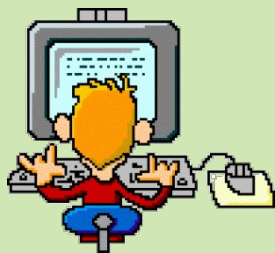
Renombrarlo como **ProgramaGenerarArbol** y realice las siguientes actividades:

▪ ACTIVIDAD 1

- a) Compilar y ejecutar
- b) Implementar el módulo **insertar** en un ABB de enteros
- c) Invocar al módulo **insertar** a partir de los elementos de la lista generada anteriormente.
- d) Invocar al módulo **imprimirpornivel** con el árbol generado en c).
- e) Graficar en papel el ABB y comprobar que los datos que se muestran en d) se corresponden con la estructura generada.



¿Qué pasaría si los valores a insertar se
presentan ordenados?



Actividades en Máquina

■ ACTIVIDAD 2

- a) Implementar un programa que invoque al módulo `crearlistaordenada` (este módulo puede armarse utilizando una modificación del algoritmo de Agregar Ordenado del programa `x.pas` - de la 1era clase) y genere un ABB con los elementos de la lista ordenada.
- b) Mostrar los datos del árbol por niveles.

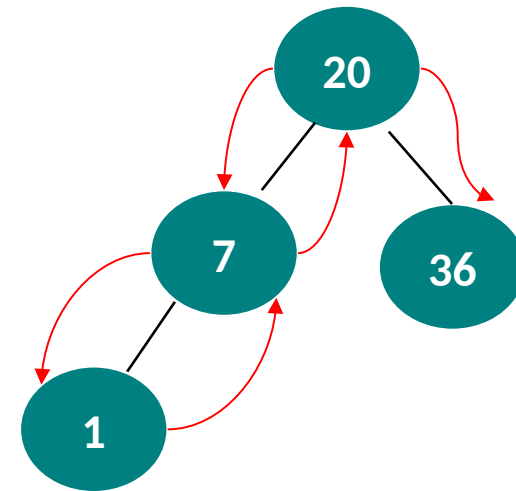
Recorridos en un árbol ABO

Los distintos recorridos permiten desplazarse a través de todos los nodos del árbol de tal forma que cada nodo sea visitado una y solo una vez.

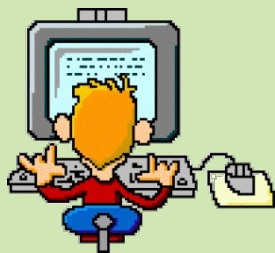
Existen varios métodos que se diferencian en el orden que se visitan:

- Recorrido Pre – Orden
- Recorrido En – Orden
- Recorrido Post – Orden

```
Procedure preOrden( a:  
arbol );  
begin  
  if ( a <> nil ) then begin  
    write (a^.dato, ' ');  
    preOrden (a^.HI);  
    preOrden (a^.HD)  
  end;  
end;
```



Imprime: 20, 7, 1, 36



Actividades en Máquina

Renombrar **ProgramaGenerarArbol** como **ProgramaArboles** y realice las siguientes actividades:

■ ACTIVIDAD 3

- a) Implementar el módulo **preOrden** que imprima los valores del ABB ya generado.
- b) Implementar el módulo **enOrden** que imprima los valores del ABB ya generado.
- c) Implementar el módulo **postOrden** que imprima los valores del ABB ya generado.
- d) Invocar cada uno de los módulos anteriores y comparar los resultados obtenidos.



Actividades en Máquina

Utilizando **ProgramaArboles** realice las siguientes actividades:

■ ACTIVIDAD 4

a) Implementar el módulo **buscar** que reciba un árbol y un valor y devuelva un puntero al nodo donde se encuentra dicho valor. En caso de no encontrarlo, debe retornar nil.

b) Invocar al módulo **buscar** con un valor que se ingresa de teclado. Mostrar el resultado de la búsqueda.



Actividades en Máquina

Utilizando **ProgramaArboles** realice las siguientes actividades:

■ ACTIVIDAD 5

- a) Implementar el módulo **verMin** que reciba un árbol y devuelva el valor mínimo. En caso de recibir un árbol vacío, retornar -1.
- b) Implementar el módulo **verMax** que reciba un árbol y devuelva el valor máximo. En caso de recibir un árbol vacío, retornar -1.
- c) Invocar a los módulos generados en a) y b). Mostrar los resultados obtenidos.