



# **TEMAS**

## **ARBOLES** **ARBOLES BINARIOS ORDENADOS** **REPRESENTACIÓN Y OPERACIONES**

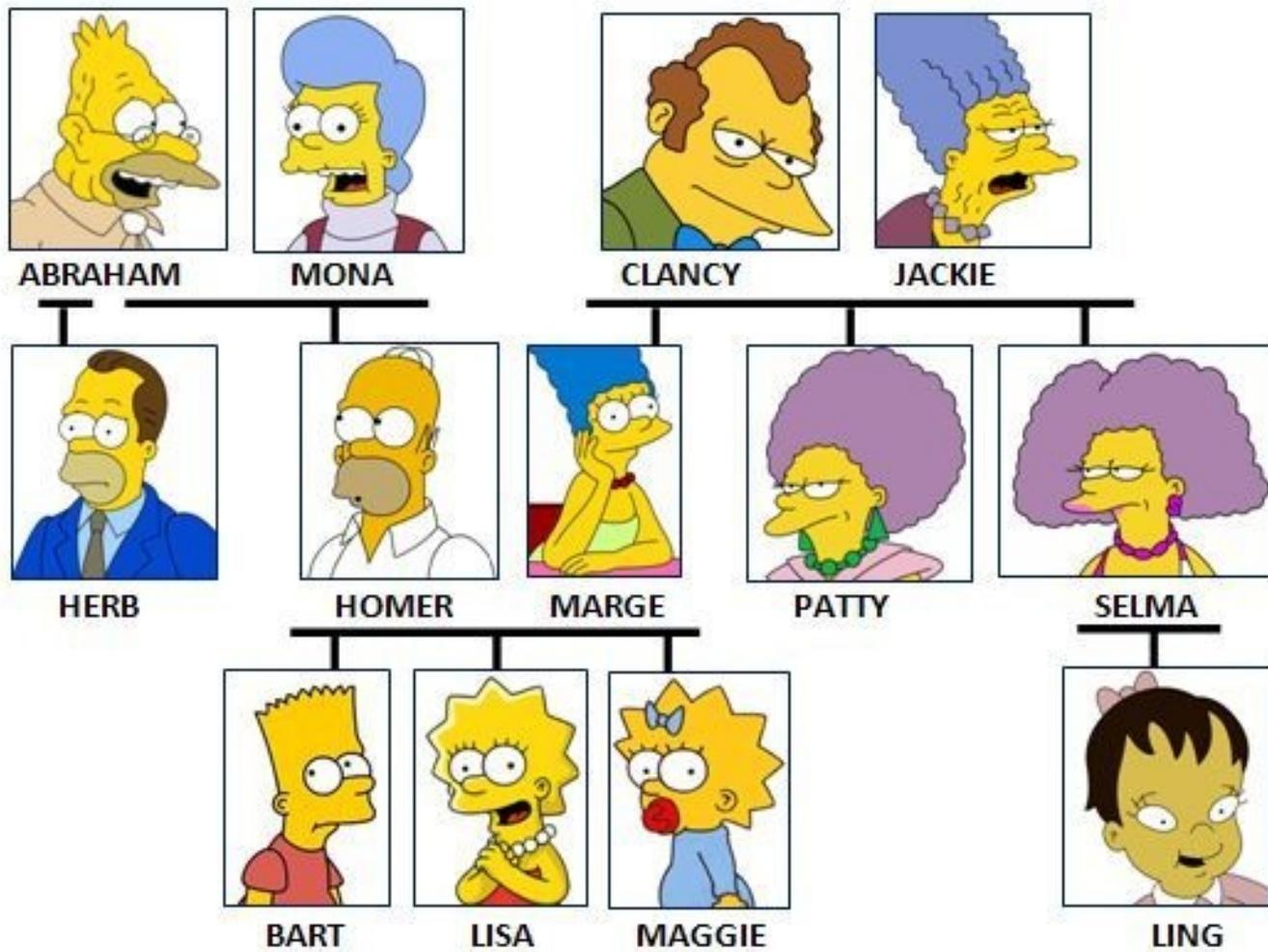
# ¿QUÉ ENTENDEMOS POR ÁRBOL?



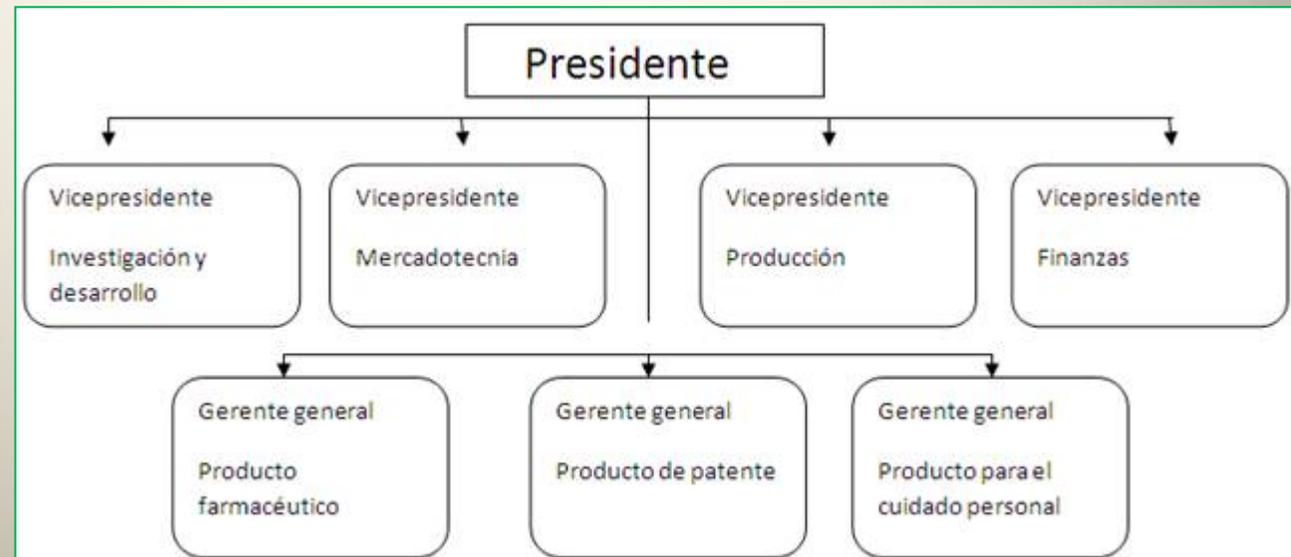
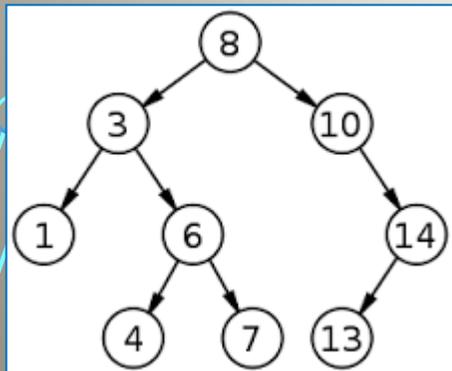
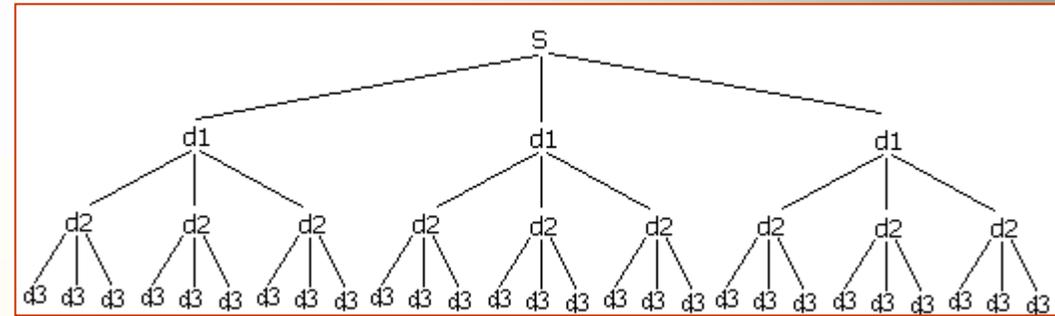
**FMK**  
foromarketing.com

# ¿QUÉ ENTENDEMOS POR ÁRBOL FAMILIAR?

## LOS SIMPSONS



# ¿QUÉ OTROS TIPOS DE ÁRBOLES CONOCEN?



# ARBOLES - CONCEPTOS



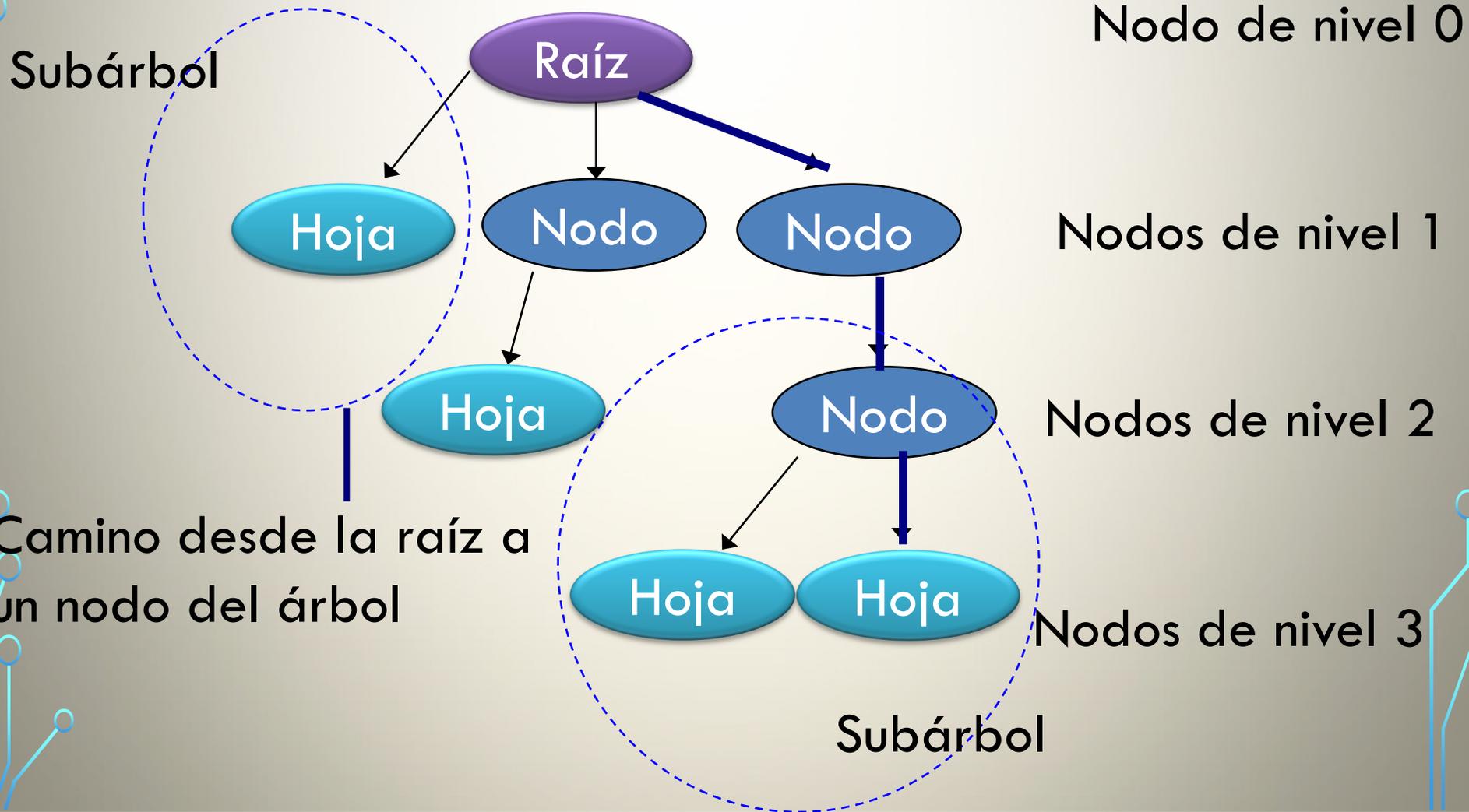
## Características

Cada elemento del árbol se relaciona con cero o más elementos a quienes llama **hijos**.

Si el árbol no está vacío, hay un único elemento al cual se llama **raíz** y que no tiene **padre** (predecesor), es decir, no es hijo de ningún otro.

Todo otro elemento del árbol posee un único padre y es un descendiente (hijo del hijo del hijo, etc.) de la raíz.

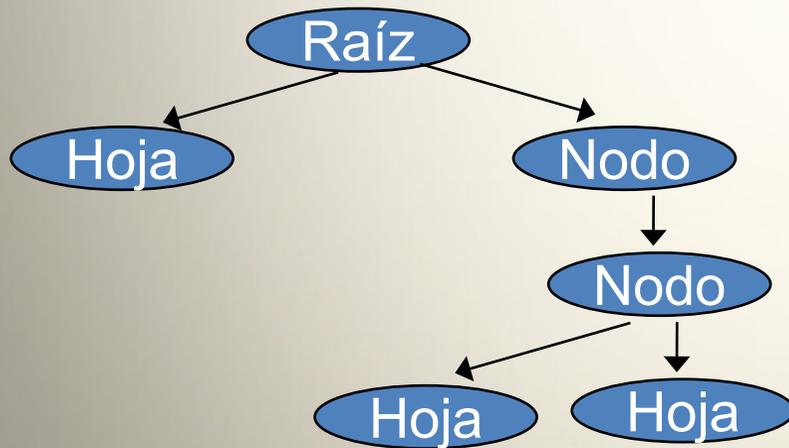
# ARBOLES - CONCEPTOS



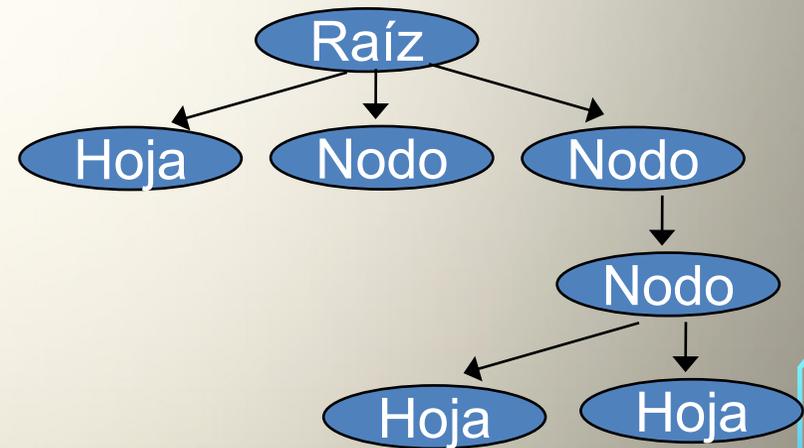
# ARBOLES - CONCEPTOS



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**



# TRABAJO GRUPAL EN EL AULA



# ¿CÓMO SE ARMA UN ÁRBOL?

## NECESITAMOS UN CRITERIO!!!



**Ejemplo:** armar el organigrama de la siguiente empresa:

Martín es el presidente de la empresa.

Paula es la vicepresidenta del área de Ventas.

Javier, Joaquín y Carolina trabajan para Paula y trabajan del departamento de marketing que pertenece al área de Ventas

Jorge es el vicepresidente de Relaciones institucionales

Pilar, Marina, Adrián y Jeremías trabajan para Jorge y son el departamento de Atención al público.

Dentro de atención al público hay una oficina donde trabajan Margarita y Mercedes para atender a los clientes extranjeros. Esta oficina depende de Marina.

¿Cómo se llama el árbol resultante según la cantidad de hijos?



## ¿CÓMO FUNCIONAN OTROS TIPOS DE ÁRBOLES?

**Supongamos que queremos agregar los siguientes números en un árbol con el siguiente criterio de edades:**

Las edades vienen desordenadas, de acuerdo a la información de abajo (El primer dato que se recibe es 55 años y siguen llegando en forma secuencial). Cuando llega la primera edad es la raíz, luego se ubican:

- a izquierda las edades más chicas que la raíz
- y a la derecha las más grandes.

Debe respetarse esta forma de agregar para cada subárbol



**55    37    72    20    65    60    67    59    28**

# ¿CÓMO FUNCIONA UN ÁRBOL BINARIO ORDENADO?



**Comparar la solución en papel en el siguiente simulador para crear un árbol binario ordenado:**

[http://aniei.org.mx/paginas/uam/CursoPoo/curso\\_poo\\_arboles.html](http://aniei.org.mx/paginas/uam/CursoPoo/curso_poo_arboles.html)



# ¿CÓMO FUNCIONA UN ARBOL BINARIO ORDENADO?



Al árbol anterior llega ahora la edad 20.

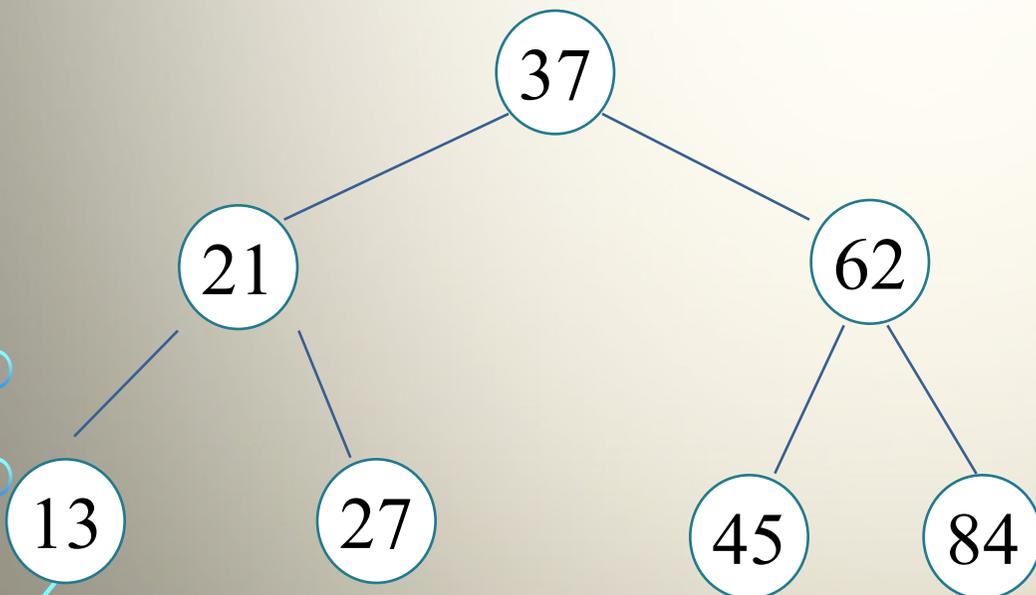
Es repetida!!

¿Qué hacemos con los repetidos?

# ARBOLES BINARIOS ORDENADOS (ABO)



Clase especial de árboles binarios en el que existe algún orden sobre los datos almacenados en ellos. Mantiene sus datos de tal manera que siempre es posible recuperarlo en el orden dado.



En este caso para cada nodo su hijo izquierdo es menor y su hijo derecho es mayor. (puede ser al revés)

# ARBOLES - CONCEPTOS



## Características

Es una estructura de datos homogénea.

Es una estructura de datos dinámica.

Es una estructura no lineal, ya que cada nodo puede tener 0,1 o más sucesores.

Es una estructura de datos jerárquica.

## ACTIVIDAD

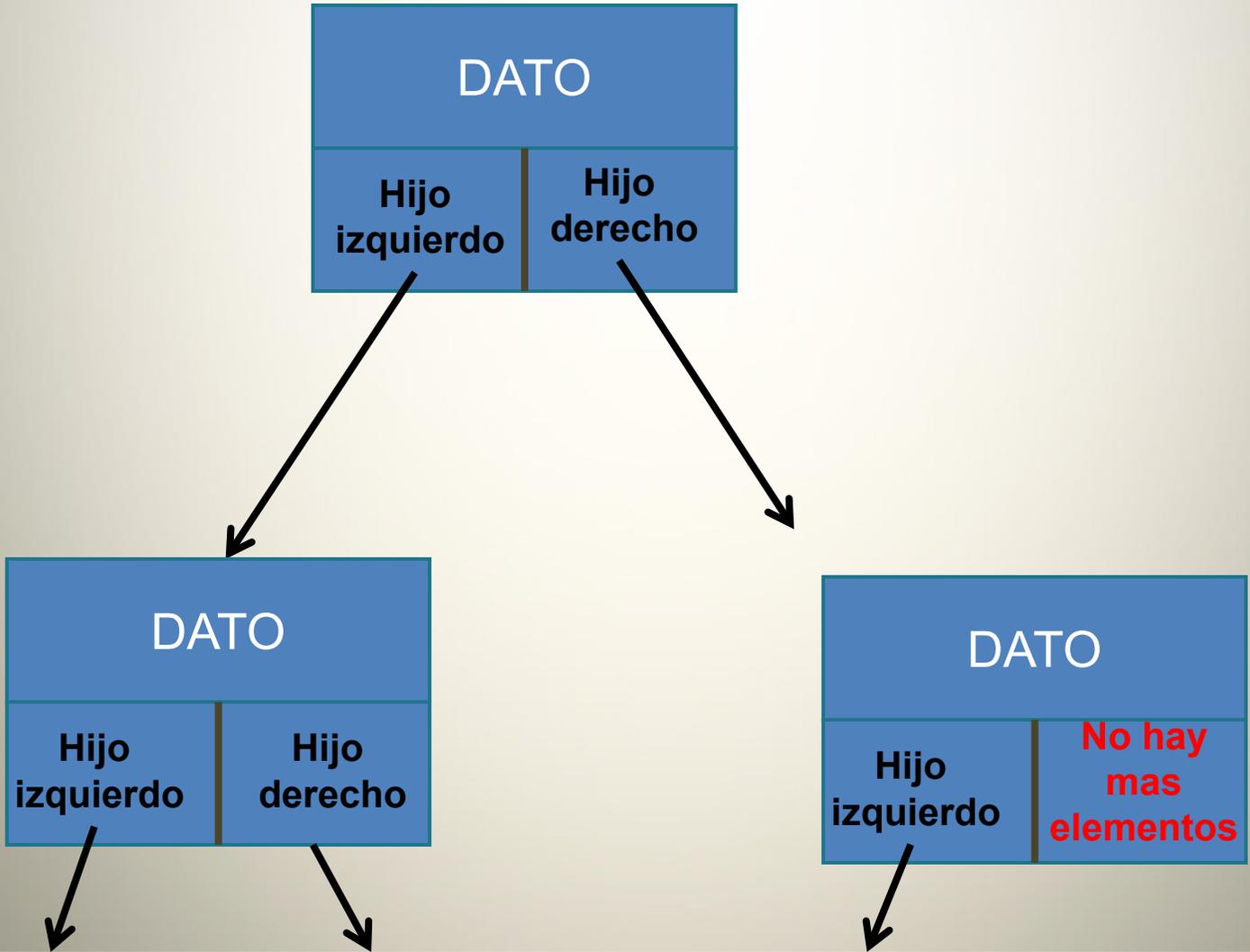


1. Dibujar un esquema del nodo de un árbol

2. Una vez acordado el tipo de nodo escribir su representación en Pascal



# ARBOLES BINARIOS NODOS Y CONEXIÓN ENTRE NODOS



} ¿cómo lo identificamos?

# ARBOLES BINARIOS- DEFINICION GENERICA



Program uno;

Type

Elemento = .....;

arbol = ^nodo;

nodo = record

hijolzq: arbol;

elem: elemento;

hijoDer: arbol;

end;

¿Con qué  
subclase de  
árboles binarios  
trabajaremos?

Char  
Integer  
Boolean  
Real  
Enumerativos  
Registros  
Listas  
Arreglos  
Arboles

# ABO- OPERACIONES



Insertar un nuevo nodo al árbol  
Mínimo y máximo  
Recorridos posibles  
Buscar un elemento  
Imprimir el contenido de un árbol  
Borrar un nodo del árbol

## ACTIVIDAD

1. Teniendo en cuenta el agregar de edades que realizamos en el árbol binario ordenado, pensar en una solución recursiva en Pascal generando un árbol binario ordenado sin elementos repetidos

Para esto contamos con el siguiente programa principal:



# ABO – OPERACIONES - AGREGAR

```
Program uno;
```

```
Type
```

```
arbol = ^nodo;
```

```
nodo = record
```

```
elem:integer;
```

```
izq:arbol
```

```
der:arbol
```

```
end;
```

```
{Definir procedimientos}
```

```
Var
```

```
a: arbol; n:integer;
```

```
Begin
```

```
inicializar(a);
```

```
read (n);
```

```
While n <> 0 do begin
```

```
agregar(a,n);
```

```
Read (n);
```

```
End;
```

```
End.
```



Agregar las  
líneas que está  
en color rojo al  
programa

# ¿CÓMO ES EL CÓDIGO EN PASCAL DE UN ALGORITMO RECURSIVO PARA AGREGAR UN NODO A UN ÁRBOL?



1. ¿Cuál o cuales son los caso base?
2. ¿Cómo se inserta un nodo, cual es el criterio?
3. ¿Donde se produce la recursión?
4. ¿Cómo se achica la solución en cada llamado?
5. ¿Cómo se resuelve en términos del mismo problema?



Una vez resueltas estas preguntas escribir el código recursivo en Pascal



# Generando código en Pascal

1. Vamos a copiar el algoritmo de **Agregar** que viene a continuación en nuestro programa, debe colocarse en la parte de procedimientos.

Notar que es recursivo



# ABO - AGREGAR



{Recurativo}

```
Procedure Agregar ( var A : arbol; n : integer);
```

```
Begin
```

```
  if A = nil Then begin { llegué al final de la rama}
```

```
    New( A );
```

```
    A^.dato := n;
```

```
    A^.izq := nil;
```

```
    A^.der := nil;
```

```
  end
```

```
else
```

```
  if n < A^.dato Then Agregar(A^.izq, n)
```

```
  else if n > A^.dato then Agregar(A^.der, n);
```

```
End;
```

Observar que si viene un repetido no lo trata y o se agrega

# ABO - AGREGAR



**Probemos con las siguientes edades.**

Considerar que los datos vienen como lo teníamos nosotros:

*36, 42, 50, 28, 26, 29, 30, 38, 31, 43*

Termina cuando llega la edad 0

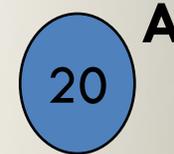
# SEGUIMIENTO DE LA EJECUCIÓN



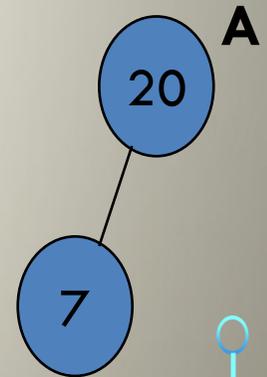
Supongamos que se lee la siguiente secuencia de números:

20 7 36 1 4 23 ¿Cómo quedará formado el árbol?

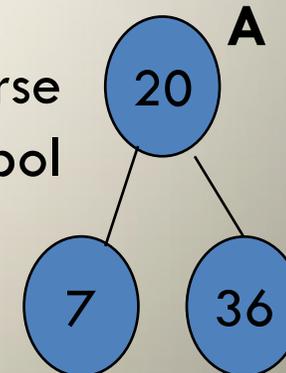
Inicialmente **A** es **nil**, por lo tanto cuando se lee el 20 queda:



Luego cuando se lee el 7 como **A** no es **nil**, debe ubicarse donde insertar, como  $7 < 20$  se toma el subárbol izquierdo, el cual como es **nil** da lugar a la inserción.



Luego cuando se lee el 36 como **A** no es **nil**, debe ubicarse donde insertar, como  $36 > 20$  se toma el subárbol derecho, el cual como es **nil** da lugar a la inserción.



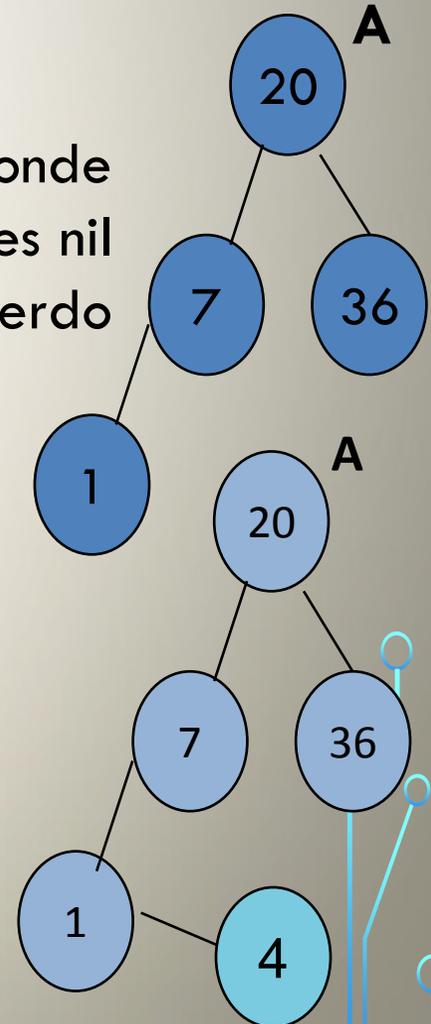
# Seguimiento de la Ejecución



20 7 36 1 4 23 ¿Cómo quedará formado el árbol?

Luego cuando se lee el 1 como A no es nil, debe ubicarse donde insertar, como  $1 < 20$  se toma el subárbol izquierdo. Como no es nil se vuelve a comparar y como  $1 < 7$  se elige el subarbol izquierdo que al ser nil da lugar a la inserción.

Luego cuando se lee el 4 como A no es nil, debe ubicarse donde insertar, como  $4 < 20$  se toma el subárbol izquierdo. Como no es nil se vuelve a comparar y dado que  $4 < 7$  se elige el subárbol izquierdo. Luego  $4 > 1$  por lo tanto se debe elegir el subárbol derecho que al ser nil da lugar a la inserción.

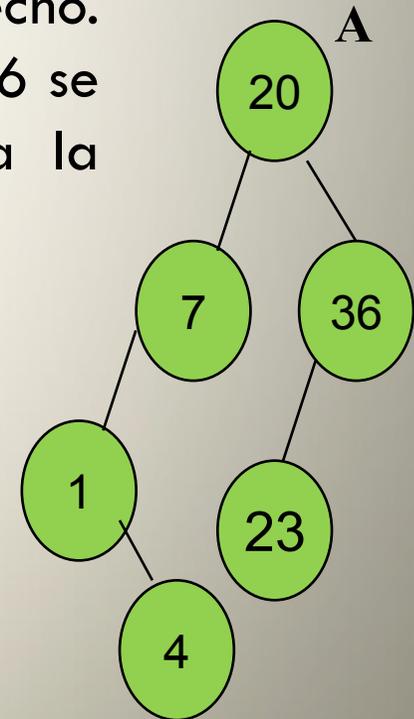


# Seguimiento de la Ejecución



20 7 36 1 4 23 ¿Cómo quedará formado el árbol?

Luego cuando se lee el 23 como A no es nil, debe ubicarse donde insertar, como  $23 > 20$  se toma el subárbol derecho. Como no es nil se vuelve a comparar y dado que  $23 < 36$  se elige el subárbol izquierdo, que al ser nil da lugar a la inserción.



¿Cómo lo implementamos?

# ABO - AGREGAR

PRECONDICION: sin repetidos!!!!



```
Procedure Agregar ( Var A : arbol;    n : elemento);
{Recurativo}
Begin
  if A = nil then begin  { llegué al final de la rama }
    New( A );
    A^.dato := n;
    A^.izq  := nil;
    A^.der  := nil;
  end
else
  if n < A^.dato then Agregar(A^.izq, n)
  else n > A^.dato then Agregar(A^.der, n);
End;
```

¿Cómo funciona la recursión

¿Cómo se realizan los enganches?

¿Qué ocurre con los repetidos?

¿Qué pasa si los valores insertados están ordenados?

# ¿Cómo mostramos los valores de un árbol?



```
{Definir procedimientos}
```

```
Var
```

```
a: arbol; n:integer;
```

```
Begin
```

```
  inicializar(a);
```

```
  read (n);
```

```
While n <> 0 do begin
```

```
  agregar(a,n);
```

```
  Read (n);
```

```
End;
```

```
Imprimir (a); End.
```

Agregar el procedimiento  
"Imprimir" al programa

# Árboles Binarios Ordenados - Operaciones



```
Procedure Imprimir ( a : arbol );  
begin  
  if ( a<> nil ) then begin  
    Imprimir (a^.izq);  
    write (a^.dato) ;  
    Imprimir (a^.der)  
  end;  
end;
```

# Árboles Binarios Ordenados – Operaciones



¿Cómo están los datos que se muestran en pantalla?

¿Cómo harías para obtener en pantalla los valores del árbol ordenados de mayor a menor?

¿Qué otros recorridos puedes hacer en el árbol?