



Escuela Profesional de
Ciencia de la Computación
Algoritmos y Estructuras de Datos
2020-B

Árboles Binarios

M.Sc. Franci Suni Lopez

Universidad Nacional de San Agustín de Arequipa

M.S. Franci Suni Lopez - UNSA

fsunilo@unsa.edu.pe

1

- Las **estructuras dinámicas** son las en la **ejecución** **varia el número de elementos y uso de memoria a lo largo del programa**.
- Entre estas tenemos:
 - Lineales (**listas** enlazadas, **pilas** y **colas**).
 - No lineales (**arboles binarios** y **grafos** o redes).

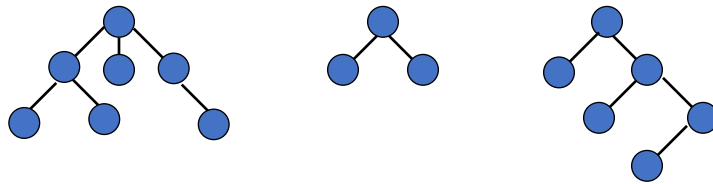
M.Sc. Franci Suni Lopez - UNSA

fsunilo@unsa.edu.pe

2

¿Qué es un Árbol?

- Es una **estructura de datos jerárquica**.
- La **relación** entre los elementos es **de uno a muchos**.



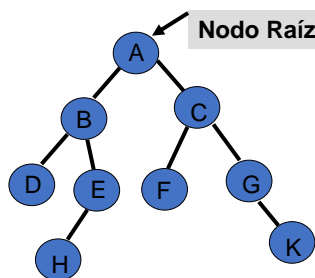
M.Sc. Franci Suni Lopez - UNSA

fsunilo@unsa.edu.pe

3

Terminología

- Nodo: Cada elemento en un árbol.
- Nodo Raíz: Primer elemento agregado al árbol.



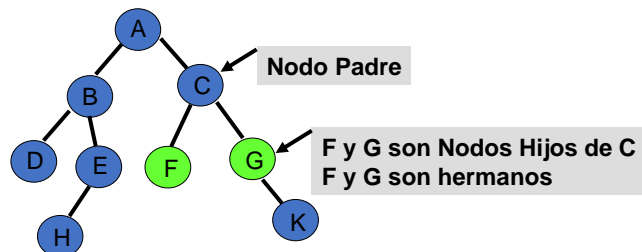
M.Sc. Franci Suni Lopez - UNSA

fsunilo@unsa.edu.pe

4

Más terminología

- Nodo Padre: Se le llama así al nodo predecesor de un elemento.
- Nodo Hijo: Es el nodo sucesor de un elemento.
- Hermanos: Nodos que tienen el mismo nodo padre.



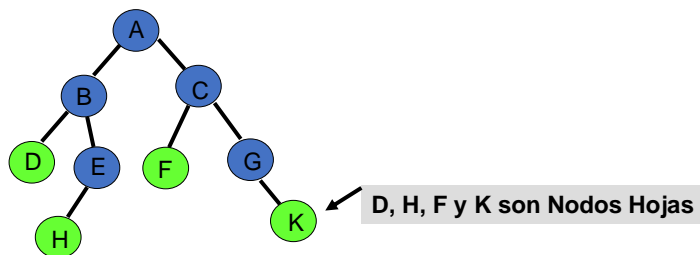
M.Sc. Franci Suni Lopez - UNSA

fsunilo@unsa.edu.pe

5

Más terminología

- Nodo Hoja: Aquel nodo que no tiene hijos.



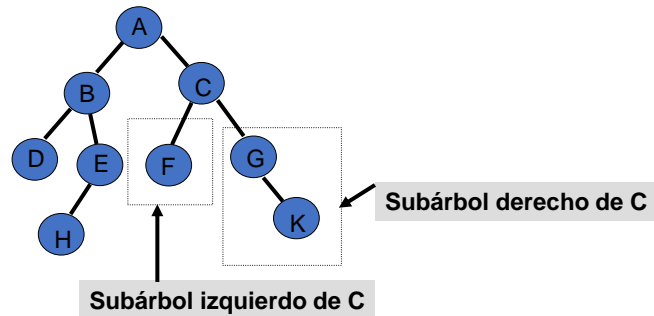
M.Sc. Franci Suni Lopez - UNSA

fsunilo@unsa.edu.pe

6

Más terminología

- Subárbol: Todos los nodos descendientes por la izquierda o derecha de un nodo.

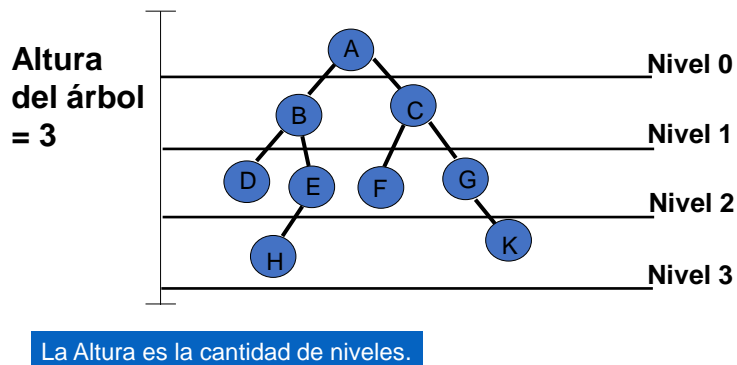


M.Sc. Franci Suni Lopez - UNSA

fsunilo@unsa.edu.pe

7

Altura y Niveles



M.Sc. Franci Suni Lopez - UNSA

fsunilo@unsa.edu.pe

8

Árbol Binario de Búsqueda (ABB)

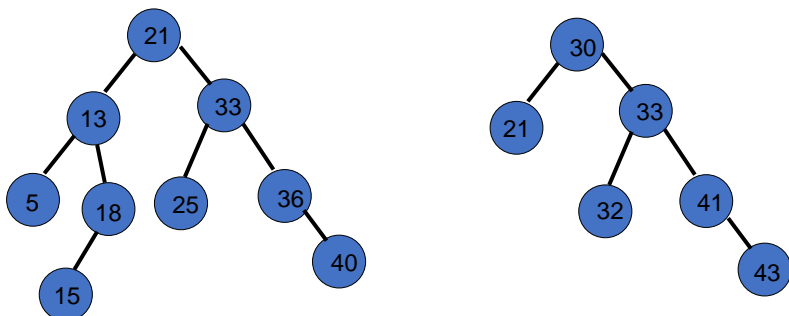
- Este tipo de árbol permite almacenar información ordenada.
- Reglas a cumplir:
 - Cada nodo del árbol puede tener 0, 1 ó 2 hijos.
 - Los descendientes *izquierdos* deben tener un valor *menor al padre*.
 - Los descendientes *derechos* deben tener un valor *mayor al padre*.

M.Sc. Franci Suni Lopez - UNSA

fsunilo@unsa.edu.pe

9

Ejemplos de ABB...

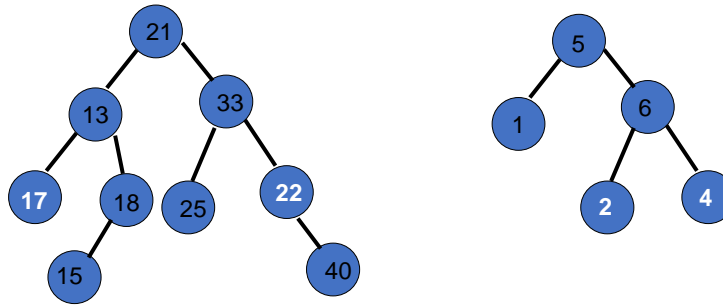


M.Sc. Franci Suni Lopez - UNSA

fsunilo@unsa.edu.pe

10

¿Por qué **no** son ABB?



M.Sc. Franci Suni Lopez - UNSA

fsunilo@unsa.edu.pe

11

Implementación de un ABB...

```

class NodoArbol
{
    public:
        T info;
        NodoArbol *izq, *der;
        NodoArbol( );
        NodoArbol(T dato);
};
NodoArbol(void) { izq = der = NULL; }
NodoArbol(T dato) { info = dato; izq = der = NULL; }
  
```

M.Sc. Franci Suni Lopez - UNSA

fsunilo@unsa.edu.pe

12

Continuación...

```

class ABB
{
    private:
        NodoArbol *raiz;
    public:
        ABB( );           // constructor
        ~ABB( );          // destructor
        //otros métodos
};

```

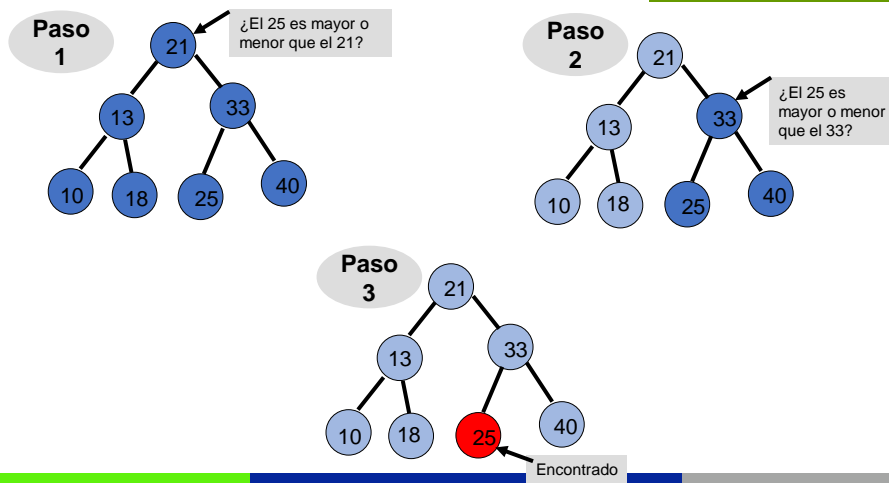
M.Sc. Franci Suni Lopez - UNSA

fsunilo@unsa.edu.pe

13

Proceso para buscar un nodo...

Buscar el 25



M.Sc. Franci Suni Lopez - UNSA

fsunilo@unsa.edu.pe

14

Implementación de la búsqueda

```

...
p=raiz;
while (p != NULL)
{ if (p->info == valor)
    return p;
  else
    p=(p->info > valor? p->izq: p->der);
}
return NULL;
...

```

← P contiene la dirección del nodo que tiene el valor buscado

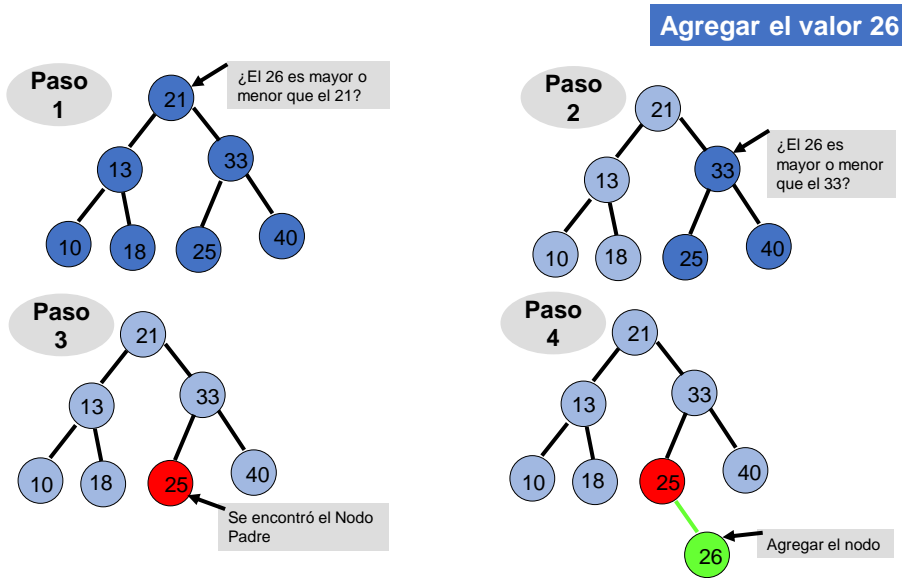
← No se encontró el valor por lo que se regresa un NULL

← Equivalente a:
 if (p -> info > valor)
 p = p -> izq;
 else p = p -> der;

Proceso para **agregar nodos...**

- Reglas:
 - El valor **a insertar no existe en el árbol.**
 - El **nuevo nodo será un Nodo Hoja** del árbol.
- Procedimiento
 1. Buscar el **Nodo Padre** del nodo a agregar.
 2. **Agregar** el nodo hoja.

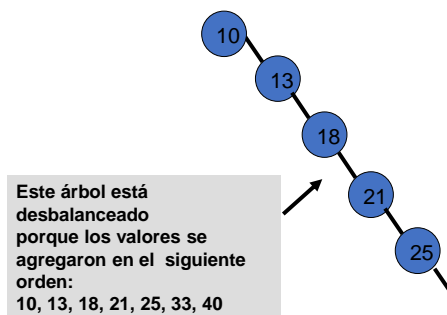
Ejemplo



17

Comentarios importantes....

- El orden de inserción de los datos, determina la forma del ABB.
- ¿Qué pasará si se insertan los datos en forma ordenada?
- La forma del ABB determina la eficiencia del proceso de búsqueda.
- Entre menos altura tenga el ABB, más balanceado estará, y más eficiente será.



Proceso para eliminar un nodo

- Si el nodo a eliminar es un:
 - Nodo hoja
 - Buscar el Nodo Padre del nodo a borrar.
 - Desconectarlo.
 - Liberar el nodo.
 - Nodo con un hijo
 - Buscar el Nodo Padre del nodo a borrar.
 - Conectar el hijo con el padre del nodo a borrar.
 - Liberar el nodo.
 - Nodo con dos hijos
 - Localizar el nodo predecesor o sucesor del nodo a borrar.
 - Copiar la información.
 - Eliminar el predecesor o sucesor según sea el caso.

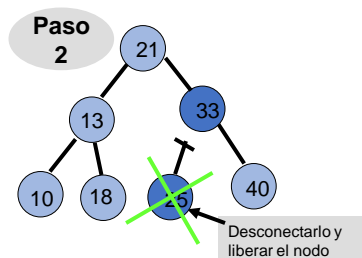
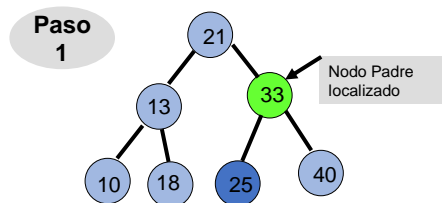
M.Sc. Franci Suni Lopez - UNSA

fsunilo@unsa.edu.pe

20

Caso: Eliminar Nodo hoja

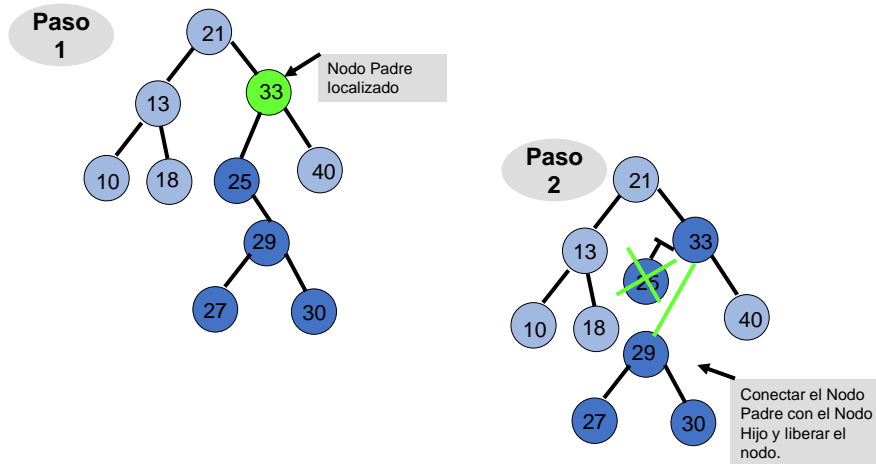
Eliminar el valor 25



21

Caso: Eliminar Nodo con un hijo

Eliminar el valor 25



22

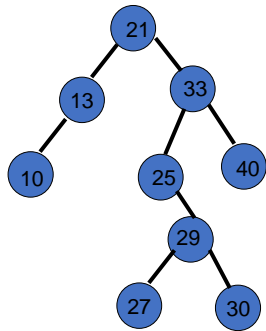
Caso: Eliminar nodo con dos hijos

1. Localizar el nodo predecesor o sucesor del nodo a borrar.
 - El PREDECESOR es "el Mayor de los Menores".
 - El SUCESOR es "el Menor de los Mayores".
 - Para la implementación es igual de eficiente programar la búsqueda del predecesor que del sucesor.
2. El valor del Predecesor (o sucesor) se copia al nodo a borrar.
3. Eliminar el nodo del predecesor (o sucesor según sea el caso).

23

Predecesor

Uno a la IZQUIERDA y todo a la DERECHA

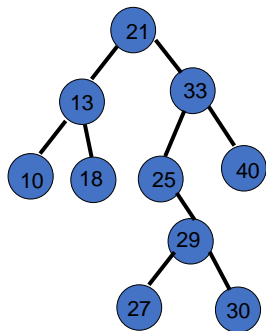


El predecesor de:	Es:
33	30
21	13
29	27

24

Sucesor

Uno a la DERECHA y todo a la IZQUIERDA



El sucesor de:	Es:
21	<input type="text"/>
33	<input type="text"/>
29	<input type="text"/>

25

Implementación del....

PREDECESOR

```
P = actual -> izq;
while( p -> der != NULL)
    p=p->der;
return p;
```

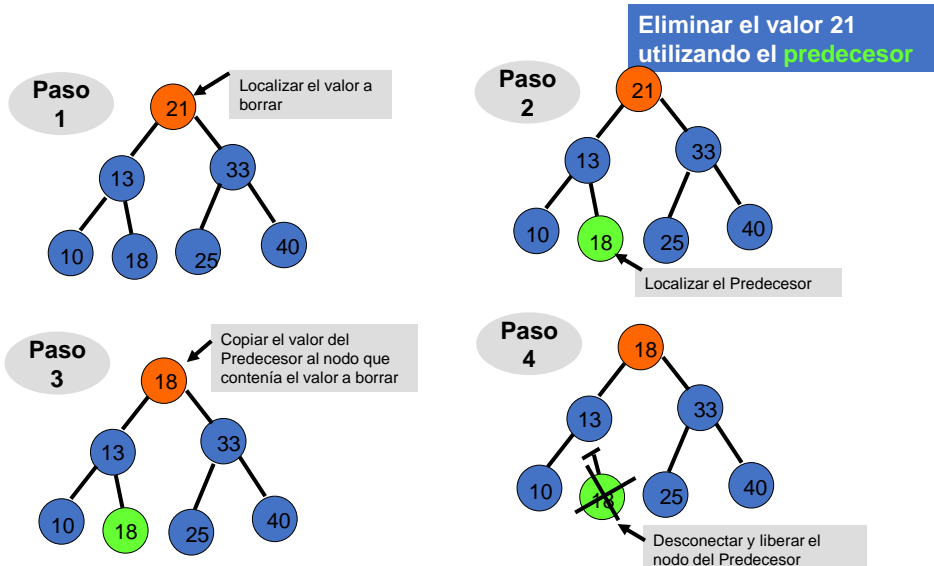
actual apunta al nodo a borrar

SUCESOR

```
P = actual -> der;
While (p -> izq != NULL )
    p=p->izq;
return p;
```

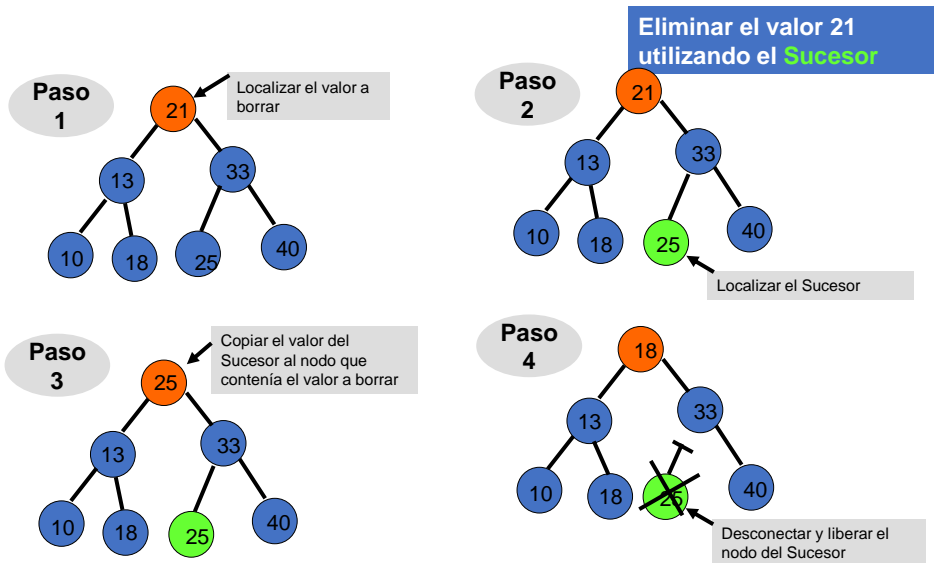
26

Caso: Eliminar Nodo con dos hijos



27

Caso: Eliminar Nodo con dos hijos



28



Thanks!

Queries?

M.Sc. Franci Suni Lopez
fsunilo@unsa.edu.pe