

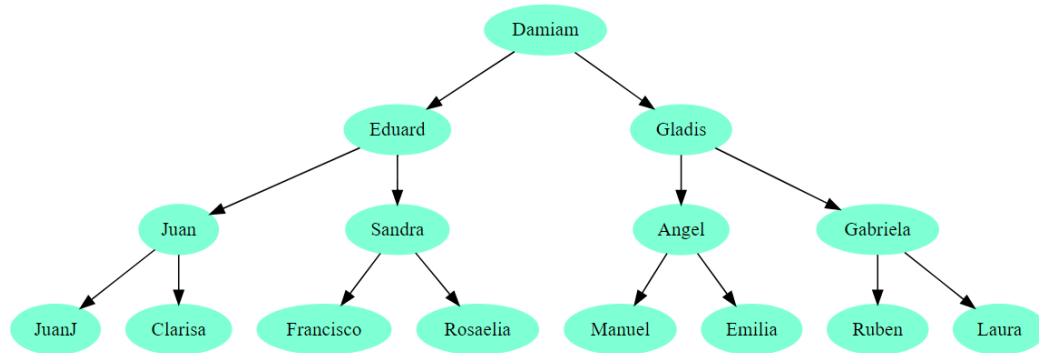
Laboratorio Nro. 5: Árboles binarios

Eduard Damiam Londoño
Universidad Eafit
Medellín, Colombia
edlondonog@eafit.edu.co

Esteban Osorio
Universidad Eafit
Medellín, Colombia
eosorio@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

1.



2. Para mejorar la eficiencia de un árbol genealógico se podría usar árbol autobalanceable como un árbol AVL o un árbol rojo-negro, pero un problema de balancear un árbol genealógico es que supondría cambios en la “descendencia” de algún nodo lo cual no sería deseable, por lo que no creo que se puede mejorar el árbol genealógico a menos de que de alguna manera (pasándolo como atributo puede ser), un nodo “recuerde” de quien es hijo o de quien es padre.
3. Primero se crea el scanner para leer las entradas y se crea el árbol para guardar los números, entonces se inicia un ciclo mientras el usuario inserte números, en el cual se irán insertando al árbol los números pasados por el usuario, cuando el usuario deja de insertar números se llama a la clase PosOrden para imprimir en árbol en PosOrden.

4.

```
public static void main(String [] args){
    Scanner input = new Scanner(System.in); // C1
    BinarySearchTree arbol = new BinarySearchTree(); //C1
    System.out.println("inserte los valores en el arbol, inserte una letra para
terminar"); //C3
    while(true){ //n
        try {
            int n= input.nextInt(); //C4
            arbol.insertar(n); //O(log n)
        }
        catch (Exception e){
            break; //C5
        }
    }
    PosOrden.posOrden(arbol); //O(n)
}
```

$O(c' + (n \cdot \log n) + n)$

$O((n \cdot \log n) + n)$

$O(n \cdot \log n)$

5. En el cálculo de complejidad anterior la variable n se refiere a la cantidad de veces que el usuario inserta un número, ahora como el tamaño del árbol es igual a la cantidad de números que inserto el usuario la variable n también se refiere al tamaño del árbol

4) Simulacro de Parcial

1. a. $altura(raiz.left)+1$
b. $altura(raiz.right)+1$
2. c
3. a. *false*;
b. 0;
c. a, $suma-a.left$;
d. a, $suma-a.right$;

4.1. c

4.2. a

4.3. d

4.4. a

5. a. `p.data == toInsert;`
b. `toInsert > p.data;`