

Laboratorio Nro. 5: Árboles binarios

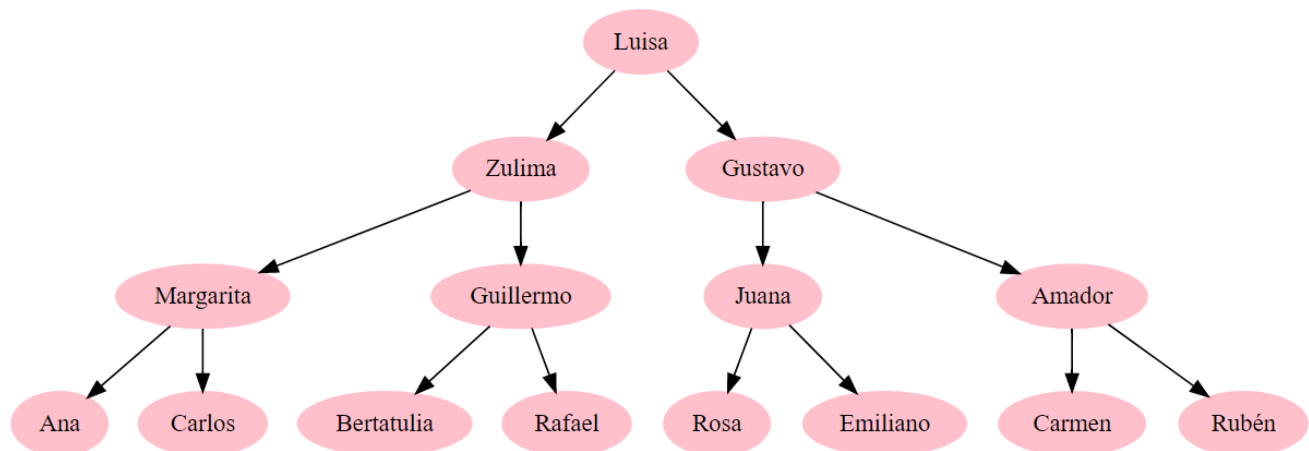
Santiago Escobar Mejía
Universidad Eafit
Medellín, Colombia
sescobarm@eafit.edu.co

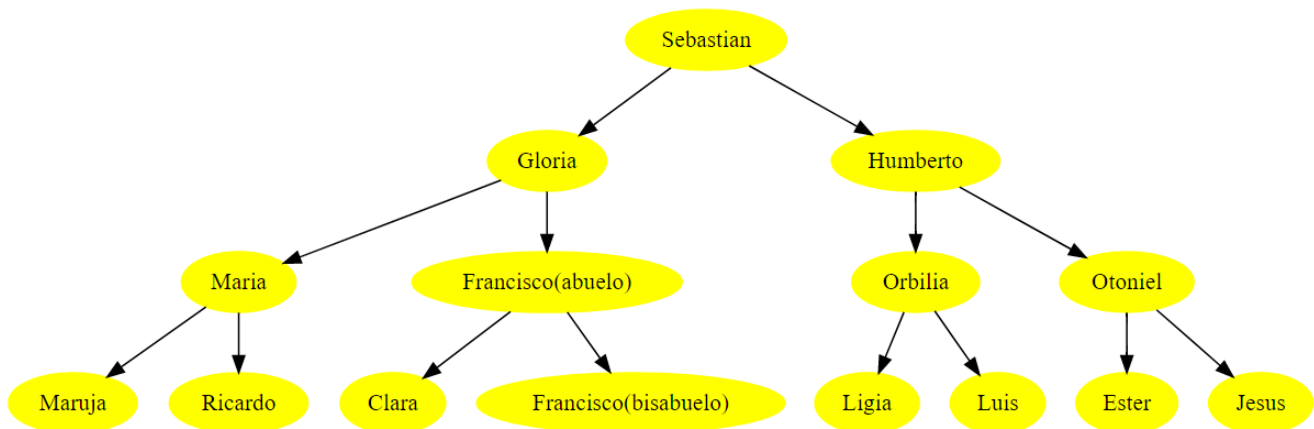
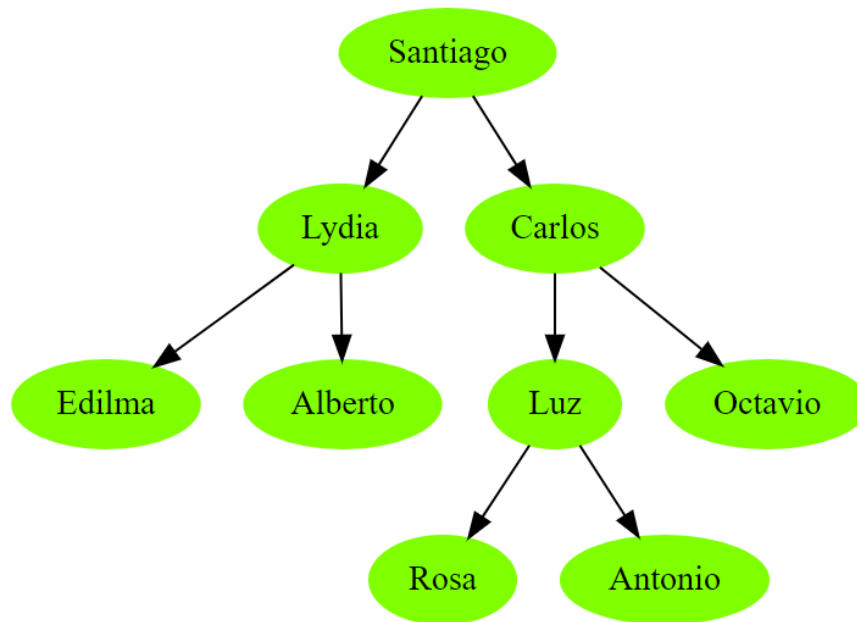
Sebastián Giraldo Gómez
Universidad Eafit
Medellín, Colombia
sgiraldog@eafit.edu.co

Luisa María Vásquez
Universidad Eafit
Medellín, Colombia
lmvasquez@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1) Investiguen cuáles son sus nombres, los nombres de sus padres, los nombres de sus abuelos y los nombres de sus bisabuelos, y construyan sus árboles genealógicos.





3.2) ¿Se puede implementar más eficientemente un árbol genealógico para que la búsqueda e inserción se puedan hacer en tiempo logarítmico? ¿O no se puede?

Para la búsqueda ,se podría reducir la complejidad si se sabe el género de la persona y el parentesco , sin embargo si se tiene solo el nombre ,la complejidad no se podría cambiar ya que el árbol no organiza los nodos por ninguna condición en específico , por lo que para agregar nodos se debe hacer de manera manual nodo por nodo, debido a que con el simple nombre de una persona no se podría saber en qué parte del árbol va .

De manera homologa , para insertar en un tiempo logarítmico se tendría que saber de antemano el parentesco para saber en que posición exacta se insertara el nodo , de lo contrario , se deberá hacer manualmente.

3.3) Expliquen con sus propias palabras cómo funcionan los ejercicios de los numerales 2.1

El ejercicio 2.1 consta de tres métodos:

1) Leer la entrada : Este método crea un árbol vacío y lee enteros desde la consola por medio del escáner y a medida que son ingresados se añaden al árbol y se repite el proceso hasta que se ingrese un número negativo que indica el final de la entrada.

2) Verificación árbol vacío : Este método se asegura de que el árbol no esté vacío , una vez confirmada la existencia de nodos en este , se pasa al siguiente método , entregándole el nodo raíz del árbol.

3) Impresión en pos orden: Para la impresión el árbol en pos orden este método verifica si hay un sub-árbol izquierdo , si lo hay , imprime su valor , de lo contrario continua a verificar si hay un sub-árbol derecho , si lo encuentra, imprime su valor , de lo contrario , va al último punto de su recorrido que es el nodo en el cual se encuentra e imprime su contenido .El proceso se repite para cada sub-árbol izquierdo o derecho.

3.4) Calculen la complejidad de los ejercicios realizados en los numerales 2.1.

```
public static void posOrden() {  
    BinarySearchTree tree=new BinarySearchTree(); //O(1)
```

```
        System.out.println("Ingrese el árbol en pre orden, al  
terminar ingrese un numero negativo");           //O(1)  
        leerElementos(tree);                       //O(n)  
        System.out.println("El recorrido del árbol en Pos-  
orden es:");                                     //O(1)  
        imprimirPosorden(tree);                   //O(n)  
  
    }
```

La complejidad del algoritmo es **O(n)** .

3.5) Expliquen con sus palabras las variables (qué es 'n', qué es 'm', etc.) del cálculo de complejidad del numeral 3.4

En la complejidad "n" representa la cantidad de elementos que son ingresados al árbol , es decir , el numero de instrucciones que debe seguir para imprimir todos los nodos del árbol.

4) Simulacro de Parcial

1)

- a) altura(raiz.izq)+1
- b) altura(raiz.der)+1

2) c) 3

3)

- a) suma==0
- b) suma==a.dato
- c) a.izq , suma-a.dato
- d) a.der , suma-a.dato

4)

- 4.1) c
- 4.2) a
- 4.3) d
- 4.4) a

5)

- a) p.dato==tolInsert
- b) tolInsert>p.dato