

## Taller en Sala 10 Árboles Binarios



**Objetivo:** 1. Explicar cómo el balanceo de árboles afecta las operaciones de un árbol binario de búsqueda. 2. Usar la notación  $O$  para encontrar formalmente la complejidad asintótica en tiempo de algoritmos



**Consideraciones:** Lean y verifiquen las consideraciones de entrega,



Trabajo en  
Parejas



Mañana,  
plazo de  
entrega



Docente entrega  
plantilla de  
código en  
GitHub



Sí .cpp, .py  
o .java



No .zip, .txt,  
html o .doc



Alumnos  
entregan  
código sin  
comprimir  
GitHub



En la carpeta Github del curso, hay **un código iniciado y un código de pruebas (tests)** que pueden explorar para solucionar los ejercicios



**Estructura del documento:** a) Datos de *vida real*, b) *Introducción* a un problema, c) Problema a resolver, d) Ayudas. Identifiquen esos elementos así:

a)



b)



c)



d)



**PhD. Mauricio Toro Bermúdez**

Docente | Escuela de Ingeniería | Informática y Sistemas  
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
Tel: (+57) (4) 261 95 00 Ext. 9473



## Ejercicios a resolver



Los árboles binarios de búsqueda se utilizan las bases de datos (por ejemplo, MySQL, Oracle, SQL Server) y en sistemas de archivos (por ejemplo, NTFS, EXT4, HFS)

1

Erradicar la pobreza en todas sus formas es el mayor desafío que enfrenta el mundo para alcanzar un desarrollo humano que no deje a nadie atrás. Un desarrollo que genere crecimiento económico, que sea socialmente responsable y ambientalmente sostenible, tal como lo establecen los Objetivos de Desarrollo Sostenible.



Colombia ha dado grandes pasos hacia la reducción de la pobreza. Entre sus logros, el porcentaje de la población que vive en pobreza multidimensional pasó de 30,4% en 2010 a 19,6% en 2018. A pesar de los avances, persiste el gran reto de cerrar las brechas de inequidad entre las regiones. Mientras que en los cascos urbanos la incidencia de la pobreza multidimensional es del 13,8% de la población, en los entornos rurales la cifra alcanza el 39,9%. Las diferencias también son notables entre mujeres y hombres, quienes no cuentan con las mismas oportunidades en el mercado laboral y, por lo tanto, no participan de manera equitativa al desarrollo económico y social del país.

Para salir de la pobreza, hacen falta iniciativas que conduzcan al crecimiento económico y a la generación de empleo y emprendimiento para los más vulnerables. Un primer paso para aportar a esas iniciativas es construir un listado telefónico. Para esto, implementen un árbol binario de búsqueda. Por simplicidad, no implementaremos el autobalanceo.



**Implementen el método buscar e insertar el nombre de una persona en situación de vulnerabilidad y su teléfono en un árbol binario de búsqueda.**  
¿Cuál es la complejidad de estos algoritmos?



**Utilicen el conjunto de datos *personas-en-situacion-de-vulnerabilidad.csv* que se encuentran en la carpeta *datasets*, en Github, para probar su algoritmo.**

**PhD. Mauricio Toro Bermúdez**

Docente | Escuela de Ingeniería | Informática y Sistemas  
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
Tel: (+57) (4) 261 95 00 Ext. 9473

**UNIVERSIDAD  
EAFIT®**

**Acreditación  
Institucional**  
Renovación  
2018 - 2026  
Resolución MEN 2158 de 2018

**2** ▶ En la clase *Arbol*, implementen un método para dibujar un árbol binario, generando código para esta herramienta: <http://www.webgraphviz.com/> ¿Cuál es la complejidad de este algoritmo?

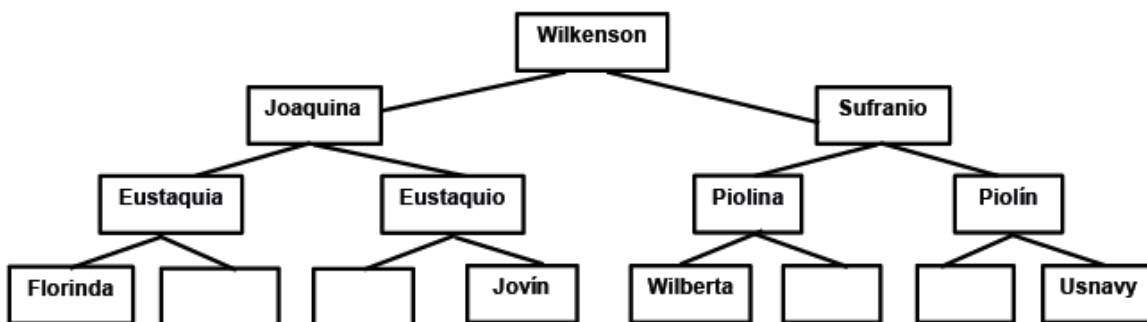
**3** ▶ **[Ejercicio Opcional]** Implementen el método borrar en un árbol binario de búsqueda. El método debe borrar un elemento en el árbol. Ideen un algoritmo que preserve la propiedad del árbol de que los mayores van a la derecha y los menores a la izquierda. ¿Cuál es la complejidad de este algoritmo?



Los árboles se utilizan en videojuegos para representar eficientemente la renderización de las escenas y para la detección de colisiones. Para ampliar la información lean en el siguiente enlace: <https://bit.ly/2KANJTs>

Como un ejemplo, vean aquí: <http://bit.ly/2x05oep>

**4** **[Ejercicio Opcional]** Implementen en Java el siguiente árbol genealógico de ancestros en la clase *Ejemplos*, en el archivo *Ejemplos.java*, dentro del método *main*



**5** **[Ejercicio Opcional]** Implementen un método para imprimir en *inorden* un árbol binario. Este método va en la clase *Arbol*.

PhD. Mauricio Toro Bermúdez

Docente | Escuela de Ingeniería | Informática y Sistemas  
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
Tel: (+57) (4) 261 95 00 Ext. 9473

UNIVERSIDAD  
**EAFIT**

Acreditación  
Institucional  
Renovación  
2018 - 2026  
Resolución MEN 2158 de 2018

# Ayudas para resolver los Ejercicios

Ejercicios 1,2,3.....	<u>Pág. 6</u>
Ejercicio 1.....	<u>Pág. 6</u>
Ejercicio 2.....	<u>Pág. 7</u>
Ejercicio 3.....	<u>Pág. 8</u>
Ejercicio 4.....	<u>Pág.8</u>
Ejercicio 5.....	<u>Pág.9</u>



## Ejercicios 1,2,3



**Pista 1:** El siguiente código es la implementación de un árbol binario de búsqueda

```
public class Node {

    public Node left ;
    public Node right ;
    public int data ;
    public Node ( int d ){
        data = d ;
    }
}

public class BinarySearchTree {

    Node root ;
    public BinarySearchTree () {
        root = null ;
    }
    public void insertar(int n) {...}
    public boolean buscar(int n) {...}
}
```



## Ejercicio 1



**Pista 1:** Implementen el siguiente método:

```
public boolean buscar(int n) {

}
```

**PhD. Mauricio Toro Bermúdez**

Docente | Escuela de Ingeniería | Informática y Sistemas  
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
Tel: (+57) (4) 261 95 00 Ext. 9473



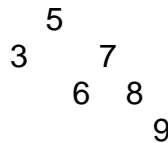


**Pista 2:** Creen un método auxiliar de la siguiente forma:

```
private boolean buscar(Node nodo, int n)
```



**Ejemplo 1:** En el árbol siguiente, está el elemento 3, pero no está el elemento 6:



## Ejercicio 1



**Pista 1:** Implementen el siguiente método:

```
public void insertar(int n) {
}

```



**Pista 2:** Creen un método auxiliar de la siguiente forma:

```
private boolean insertar(Node nodo, int n)
```



**Pista 3:** En una hoja de papel, inserten en un árbol binario ordenado los números del 1 al 10 en este orden:

- 1,2,3,4,5,6,7,8,9,10
- 5,3,6,2,7,1,9,8,10

**PhD. Mauricio Toro Bermúdez**

Docente | Escuela de Ingeniería | Informática y Sistemas  
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
Tel: (+57) (4) 261 95 00 Ext. 9473





**Pista 4:** Herramientas como BlueJ y Jgrasp les permiten observar gráficamente el árbol



### Ejercicio 3



**Pista 1:** Implementen el siguiente método:

```
public void borrar(int n) {  
  
}
```



**Pista 2:** Creen un método auxiliar de la siguiente forma:

```
private boolean borrar(Node nodo, int n)
```



### Ejercicio 4



**Error Común:**

**PhD. Mauricio Toro Bermúdez**

Docente | Escuela de Ingeniería | Informática y Sistemas  
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
Tel: (+57) (4) 261 95 00 Ext. 9473





## Ejercicio 5



**Pista 1:** Este método se usa en el método *main* de la clase *Ejemplos*.

**PhD. Mauricio Toro Bermúdez**

Docente | Escuela de Ingeniería | Informática y Sistemas  
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
Tel: (+57) (4) 261 95 00 Ext. 9473





# ¿Alguna inquietud?

## CONTACTO

Docente Mauricio Toro Bermúdez

Teléfono: (+57) (4) 261 95 00 Ext. 9473

Correo: [mtorobe@eafit.edu.co](mailto:mtorobe@eafit.edu.co)

Oficina: 19- 627

Agenden una cita dando clic en la pestaña

-*Semana*- de <http://bit.ly/2gzVg10>