

Informe de Laboratorio 05

Tema: Arbol AVL

Nota

Estudiante	Escuela	Asignatura
Misael Marrón Lope mmarronl@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	EDA Semestre: III Código: 20220575

Laboratorio	Tema	Duración
05	Arbol AVL	

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - A	2023	27 junio 2023

1. Tarea

- Implementa una clase de arbol AVL.
- Implemente una clase Node donde T es un tipo genérico, esta clase debe contener al menos dos propiedades.
- Utilizar Git para evidenciar su trabajo.
- Enviar trabajo al profesor en un repositorio GitHub Privado, dándole permisos como colaborador.

2. Equipos, materiales y temas utilizados

- Sistema Operativo Windows 10 ver. 22H2
- Eclipse IDE, Visual studio
- java 20.0.1
- Git 2.40.1.
- Cuenta en GitHub con el correo institucional.

3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- <https://github.com/MisaelMarron/eda-lab-b-23a.git>
- URL para el laboratorio 05 en el Repositorio GitHub.
- <https://github.com/MisaelMarron/eda-lab-b-23a/tree/main/lab05>

4. Actividades : La creacion de Arbol Binario de Busqueda

- Elabore un informe implementando Arboles AVL con toda la lista de operaciones search(), getMin(), getMax(), parent(), son(), insert(), remove() rotateLeft(), rotateRight().

4.1. Commits Importantes:

Listing 1: Mi primer commit importante es cuando agregue la clase Nodo .

```
commit 2d1804f22c372b0bb3cd890f63a6964a2dfa029e
Author: Misael Marron <mmarronl@unsa.edu.pe>
Date: Mon Jun 26 00:14:54 2023 -0500
```

Agregamos clase Nodo<T>

```
7
8 class Nodo<T extends Comparable<T>> {
9     T Clave;
10    int cont;
11    int Nivel;
12    Nodo<T> izq, der;
13
14    Nodo(T value) {
15        Clave = value;
16        cont = 1;
17        Nivel = 1;
18    }
19    public T getClave() {
20        return Clave;
21    }
22 }
23
```

Listing 2: Mi segundo commit mas importante es cuando agregue el metodo insert .

```
commit 172dd67df73cc425f2f2afca7cc30e83d35e6689
Author: Misael Marron <mmarronl@unsa.edu.pe>
Date: Mon Jun 26 01:07:39 2023 -0500
```

Agregamos metodo insert

```
45 public void insert(T Clave) {
46     root = insertNode(root, Clave);
47 }
48
49 private Nodo<T> insertNode(Nodo<T> node, T Clave) {
50     if (node == null)
51         return new Nodo<>(Clave);
52     int cmp = Clave.compareTo(node.Clave);
53     if (cmp < 0)
54         node.izq = insertNode(node.izq, Clave);
55     else if (cmp > 0)
56         node.der = insertNode(node.der, Clave);
57     else
58         node.cont++;
59     node.Nivel = 1 + Math.max(Nivel(node.izq), Nivel(node.der));
60     int balance = balanceFactor(node);
61     if (balance > 1 && Clave.compareTo(node.izq.Clave) < 0)
62         return rotateder(node);
63     if (balance < -1 && Clave.compareTo(node.der.Clave) > 0)
64         return rotateizq(node);
65     if (balance > 1 && Clave.compareTo(node.izq.Clave) > 0) {
66         node.izq = rotateizq(node.izq);
67         return rotateder(node);
68     }
69     if (balance < -1 && Clave.compareTo(node.der.Clave) < 0) {
70         node.der = rotateder(node.der);
71         return rotateizq(node);
72     }
73     return node;
74 }
75
```

Listing 3: Mi tercer commit es cuando agregue el max y min y los metodos para rotar

```
commit 3a2a47fb461e4a3a39434fb2f07cc4d8990fc2fa
```

```
Author: Misael Marron <mmarronl@unsa.edu.pe>
```

```
Date: Mon Jun 26 01:30:24 2023 -0500
```

Agregamos max y min

```
132 private Nodo<T> rotateder(Nodo<T> y) {  
133     Nodo<T> x = y.izq;  
134     Nodo<T> T2 = x.der;  
135     x.der = y;  
136     y.izq = T2;  
137     y.Nivel = Math.max(Nivel(y.izq), Nivel(y.der)) + 1;  
138     x.Nivel = Math.max(Nivel(x.izq), Nivel(x.der)) + 1;  
139     return x;  
140 }  
141  
142 private Nodo<T> rotateizq(Nodo<T> x) {  
143     Nodo<T> y = x.der;  
144     Nodo<T> T2 = y.izq;  
145     y.izq = x;  
146     x.der = T2;  
147     x.Nivel = Math.max(Nivel(x.izq), Nivel(x.der)) + 1;  
148     y.Nivel = Math.max(Nivel(y.izq), Nivel(y.der)) + 1;  
149     return y;  
150 }
```

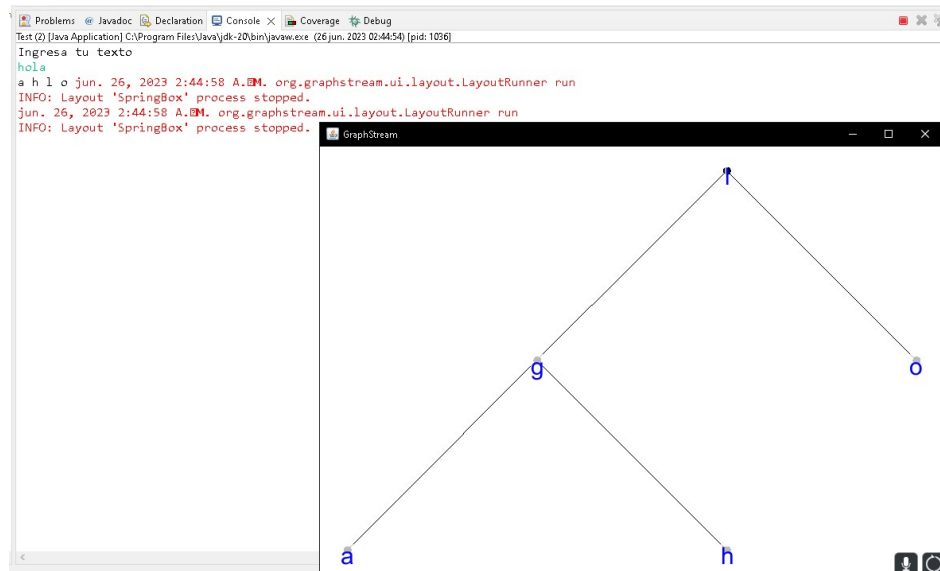
Listing 4: Mi cuarto y ultimo commit es cuando termine y agregue parte grafica y test

commit 39b92fdc8323be3073a8540f9243d992d6fbc794
Author: Misael Marron <mmarronl@unsa.edu.pe>
Date: Mon Jun 26 02:32:16 2023 -0500

Clase prueba

```
273 public void imprimirArbol() {
274     Graph graph = new SingleGraph("Árbol AVL");
275     agregarNodos(graph, root, 0, 0);
276     agregarAristas(graph, root);
277     System.setProperty("org.graphstream.ui", "swing");
278     Viewer viewer = graph.display();
279     viewer.disableAutolayout();
280     viewer.setCloseFramePolicy(Viewer.CloseFramePolicy.CLOSE_VIEWER);
281 }
282
283 public void cerrarArbol(){
284     Graph graph = new SingleGraph("Árbol AVL");
285     agregarNodos(graph, root, 0, 0);
286     agregarAristas(graph, root);
287     System.setProperty("org.graphstream.ui", "swing");
288     Viewer viewer = graph.display();
289     viewer.disableAutolayout();
290     viewer.setCloseFramePolicy(Viewer.CloseFramePolicy.EXIT);
291 }
292
293 private void agregarNodos(Graph graph, Nodo<T> nodo, double x, double y) {
294     if (nodo != null) {
295         Node node = graph.addNode(nodo.Clave.toString());
296         node.setAttribute("ui.label", nodo.Clave.toString());
297         node.setAttribute("ui.style", "text-size: 30px; text-alignment: center; text-background-mode: rounded-box; text-background-color: #ffffffbb;");
298         node.setAttribute("xy", x, y);
299         agregarNodos(graph, nodo.izq, x - 4.0, y - 4.0);
300         agregarNodos(graph, nodo.der, x + 4.0, y - 4.0);
301     }
302 }
303
304 private void agregarAristas(Graph graph, Nodo<T> nodo) {
305     if (nodo != null) {
306         if (nodo.izq != null) {
307             Edge edge = graph.addEdge(nodo.Clave + "-" + nodo.izq.Clave, nodo.Clave.toString(), nodo.izq.Clave.toString());
308             edge.setAttribute("ui.style", "fill-color: black;");
309         }
310         if (nodo.der != null) {
311             Edge edge = graph.addEdge(nodo.Clave + "-" + nodo.der.Clave, nodo.Clave.toString(), nodo.der.Clave.toString());
312             edge.setAttribute("ui.style", "fill-color: black;");
313         }
314     }
315 }
```

```
1 import java.util.List;
2 import java.util.Scanner;
3
4
5 public class Test {
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8         AVL<Character> arbol = new AVL<>();
9
10        System.out.println("Ingresa tu texto");
11        String Prueba = scanner.nextLine();
12
13        arbol = input(arbol, Prueba);
14        arbol.inOrderTraversal();
15
16        char character;
17        List<Character> childs = null;
18
19        System.setProperty("org.graphstream.ui", "graphstream.ui.swing");
20        arbol.imprimirArbol();
21
22        arbol.insert('a');
23        arbol.insert('g');
24        arbol.imprimirArbol();
25    }
26
27    public static AVL<Character> input(AVL<Character> arbolBinario, String Palabra){
28        for (int i = 0; i < Palabra.length(); i++) {
29            arbolBinario.insert(Palabra.charAt(i));
30        }
31        return arbolBinario;
32    }
33
34 }
35
```



4.2. Estructura de laboratorio 05

- El contenido que se entrega en este laboratorio es el siguiente:

```
lab05/
|--- codigo
|   |--- test.java
|   |--- AVL.java
|--- latex
|   |--- img
|       |--- logo_abet.png
|       |--- logo_episunsa.png
|       |--- logo_unsa.jpg
|       |--- codigo1.jpg
|       |--- codigo2.jpg
|       |--- codigo2.jpg
|       |--- codigo3.jpg
|       |--- codigo4.jpg
|       |--- codigo4b.jpg
|       |--- ejecucion.jpg
|--- Lab05-MisaelMarron.pdf
|--- Lab05-MisaelMarron.tex
```

5. Preguntas:

- ¿Explique como es el algoritmo que implemento para obtener el arbol binario de busqueda con la libreria Graph Stream? Recuerde que pueden haber operaciones sobre el BST.
- El algoritmo para mostrar un árbol binario de búsqueda con Graph Stream consta de los siguientes pasos:
Crear un objeto Graph para representar el grafo. Definir las características visuales del grafo. Agregar nodos y aristas al grafo recursivamente. Mostrar el grafo en una ventana utilizando el visor de Graph Stream.

6. Rúbricas

6.1. Entregable Informe

Tabla 1: Tipo de Informe

Informe	
Latex	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y facil de leer.

6.2. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumple con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos los ítems.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Tabla 2: Niveles de desempeño

	Nivel			
Puntos	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

Tabla 3: Rúbrica para contenido del Informe y demostración

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	3	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	2	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	1.5	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	1	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2	X	2	
7. Ortografía	El documento no muestra errores ortográficos.	2	X	1.5	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	4	
Total		20		17	

7. Referencias

- <https://docs.oracle.com/javase/tutorial/java/generics/types.html>
- <https://www.eclipse.org/downloads/packages/release/2022-03/r/eclipse-ide-enterprise-java-and-webtools>
- <https://www.w3schools.com/java/default.asp>