

## Clase Nodo

The screenshot shows the IntelliJ IDEA interface with the following details:

- File Bar:** D, Version control, Current File, Trial.
- Project Bar:** C:\Users\lizet\ideaProjects\ArbolesED
- Editor:** The Nodo.java file is the active tab, containing the following code:

```
1 public class Nodo { 1 usage
2     String data; 1 usage
3     Nodo left, right; 1 usage
4
5     public Nodo(String data) { no usages
6         this.data = data;
7         this.left = null;
8         this.right = null;
9     }
10}
11
```
- Sidebar:** Shows the project structure with files like Main.java, Arbol.java, and PruebaArbol.java.
- Bottom Bar:** Shows the command-line arguments for the Java runtime environment.

## Clase Árbol

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project:** ArbolesED (C:\Users\lizet\IdeaProjects\ArbolesED)
- Current File:** Arbol.java
- Code Content:** The code defines a binary search tree class named Arbol. It includes methods for inserting values into the tree and searching for specific values.

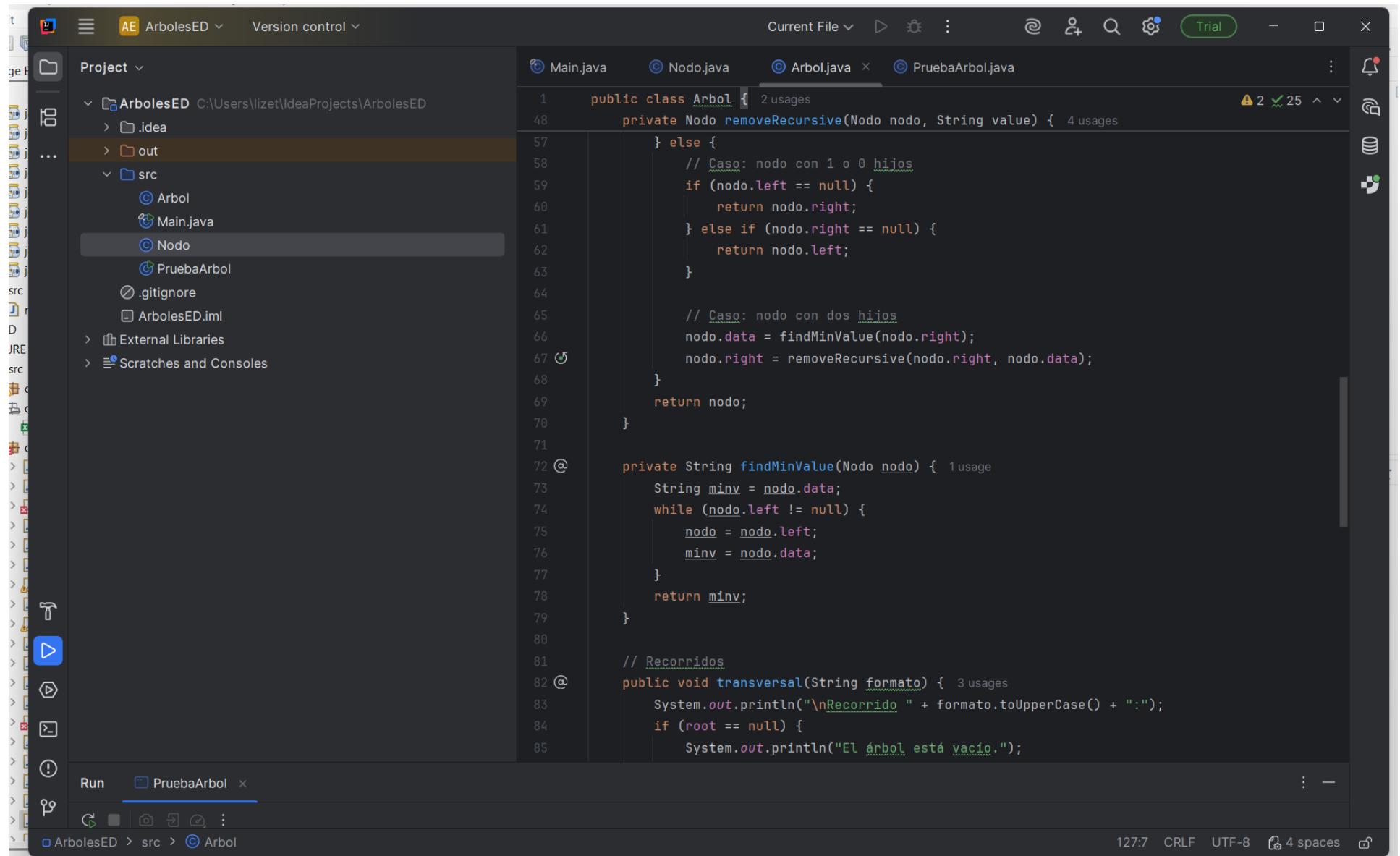
```
1 public class Arbol { 2     private Nodo root; 3 4     public Arbol() { 5         this.root = null; 6     } 7 8     // Insertar un nodo 9     public void insert(String value) { 10         root = insertRecursive(root, value); 11     } 12 13     private Nodo insertRecursive(Nodo nodo, String value) { 14         if (nodo == null) { 15             return new Nodo(value); 16         } 17 18         if (value.compareTo(nodo.data) < 0) { 19             nodo.left = insertRecursive(nodo.left, value); 20         } else if (value.compareTo(nodo.data) > 0) { 21             nodo.right = insertRecursive(nodo.right, value); 22         } 23         return nodo; 24     } 25 26     // Buscar un nodo 27     public Nodo search(String value) { 28         return searchRecursive(root, value); 29     } 30 31     private Nodo searchRecursive(Nodo nodo, String value) { 32         if (nodo == null) { 33             return null; 34         } 35         if (value.compareTo(nodo.data) < 0) { 36             return searchRecursive(nodo.left, value); 37         } else if (value.compareTo(nodo.data) > 0) { 38             return searchRecursive(nodo.right, value); 39         } 40         return nodo; 41     } }
```

- Run Configuration:** PruebaArbol
- Status Bar:** 127:7 CRLF UTF-8 4 spaces

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project View:** Shows the project structure under "ArbolesED". The "src" folder contains "Arbol", "Main.java", "Nodo", and "PruebaArbol".
- Code Editor:** The "Arbol.java" file is open. The code implements a binary search tree with methods for insertion, search, and removal.
- Toolbars:** Standard IntelliJ IDEA toolbars for navigation, search, and file operations are visible at the top.
- Status Bar:** Shows the file path "ArbolesED > src > Arbol", line count "127:7", encoding "CRLF", character set "UTF-8", and code style settings "4 spaces".

```
1 public class Arbol { 2     } 3 4     private Nodo searchRecursive(Nodo nodo, String value) { 5         if (nodo == null || nodo.data.equals(value)) { 6             return nodo; 7         } 8 9         if (value.compareTo(nodo.data) < 0) { 10             return searchRecursive(nodo.left, value); 11         } else { 12             return searchRecursive(nodo.right, value); 13         } 14     } 15 16     // Eliminar un nodo 17     public void remove(String value) { 18         root = removeRecursive(root, value); 19     } 20 21     private Nodo removeRecursive(Nodo nodo, String value) { 22         if (nodo == null) { 23             return null; 24         } 25 26         if (value.compareTo(nodo.data) < 0) { 27             nodo.left = removeRecursive(nodo.left, value); 28         } else if (value.compareTo(nodo.data) > 0) { 29             nodo.right = removeRecursive(nodo.right, value); 30         } else { 31             // Caso: nodo con 1 o 0 hijos 32         } 33     } 34 }
```



The screenshot shows the IntelliJ IDEA interface with the following details:

- Project:** ArbolesED (C:\Users\lizet\IdeaProjects\ArbolesED)
- Current File:** Arbol.java
- Code Preview:**

```
1  public class Arbol { 2 usages
82     public void transversal(String formato) { 3 usages
84         if (root == null) {
85             System.out.println("El árbol está vacío.");
86             return;
87         }
88
89         switch (formato.toLowerCase()) {
90             case "inorden":
91                 recorridoIn(root);
92                 break;
93             case "preorden":
94                 recorridoPre(root);
95                 break;
96             case "postorden":
97                 recorridoPos(root);
98                 break;
99             default:
100                 System.out.println("Formato inválido (usa inorden, preorder o postorden).");
101             }
102             System.out.println();
103         }
104
105         private void recorridoIn(Nodo nodo) { 3 usages
106             if (nodo != null) {
107                 recorridoIn(nodo.left);
108                 System.out.print(nodo.data + " ");
109                 recorridoIn(nodo.right);
110             }
111         }
112     }
```
- Status Bar:** 127:7 CRLF UTF-8 4 spaces

The screenshot shows a Java project named "ArbolesED" in an IDE. The project structure is as follows:

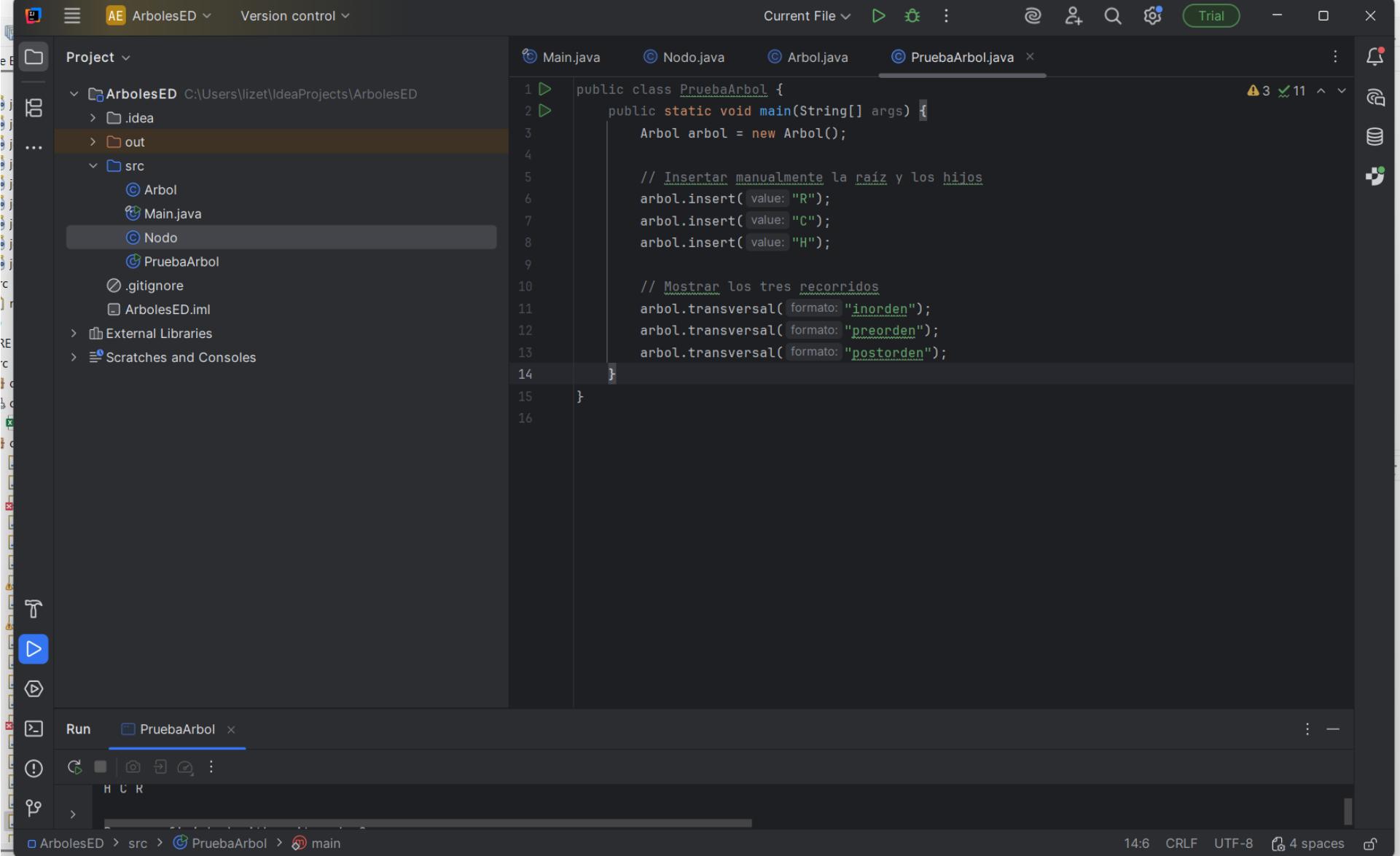
- Project: ArbolesED
- src:
  - Arbol
  - Main.java
  - Nodo
  - PruebaArbol
- .gitignore
- ArbolesED.iml
- External Libraries
- Scratches and Consoles

The file "Arbol.java" is open in the editor, displaying the following code:

```
1 public class Arbol { 2 usages  
104  
105     private void recorridoIn(Nodo nodo) { 3 usages  
106         if (nodo != null) {  
107             recorridoIn(nodo.left);  
108             System.out.print(nodo.data + " ");  
109             recorridoIn(nodo.right);  
110         }  
111     }  
112  
113     private void recorridoPre(Nodo nodo) { 3 usages  
114         if (nodo != null) {  
115             System.out.print(nodo.data + " ");  
116             recorridoPre(nodo.left);  
117             recorridoPre(nodo.right);  
118         }  
119     }  
120  
121     private void recorridoPos(Nodo nodo) { 3 usages  
122         if (nodo != null) {  
123             recorridoPos(nodo.left);  
124             recorridoPos(nodo.right);  
125             System.out.print(nodo.data + " ");  
126         }  
127     }  
128 }
```

The "Run" tab in the bottom navigation bar is selected, showing "PruebaArbol". The status bar at the bottom right indicates 127:7 CRLF UTF-8 4 spaces.

## Clase Prueba



The screenshot shows the IntelliJ IDEA interface with the following details:

- Project:** ArbolesED (C:\Users\lizet\deaProjects\ArbolesED)
- Files:** Main.java, Nodo.java, Arbol.java, PruebaArbol.java
- PruebaArbol.java Content:**

```
1 public class PruebaArbol {
2     public static void main(String[] args) {
3         Arbol arbol = new Arbol();
4
5         // Insertar manualmente la raiz y los hijos
6         arbol.insert(value: "R");
7         arbol.insert(value: "C");
8         arbol.insert(value: "H");
9
10        // Mostrar los tres recorridos
11        arbol.transversal(formato: "inorden");
12        arbol.transversal(formato: "preorden");
13        arbol.transversal(formato: "postorden");
14    }
15}
```

- Run Tab:** PruebaArbol
- Status Bar:** 14:6 CRLF UTF-8 4 spaces

## Consola

The screenshot shows the IntelliJ IDEA interface with a Java project named "ArbolesED". The project structure on the left includes files like Main.java, Nodo.java, Arbol.java, and PruebaArbol.java. The "PruebaArbol.java" file is open in the editor, containing code to insert values into a tree and perform three traversal methods: in-order, pre-order, and post-order. The "Run" tab in the bottom-left shows the execution of "PruebaArbol". The terminal output on the right displays the results of these traversals: in-order (C H R), pre-order (R C H), and post-order (H C R). The status bar at the bottom indicates the current file is "main".

```
public class PruebaArbol {
    public static void main(String[] args) {
        Arbol arbol = new Arbol();

        // Insertar manualmente la raiz y los hijos
        arbol.insert( value: "R");
        arbol.insert( value: "C");
        arbol.insert( value: "H");

        // Mostrar los tres recorridos
        arbol.transversal( formato: "inorden");
        arbol.transversal( formato: "preorden");
        arbol.transversal( formato: "postorden");
    }
}

C:\Users\lizet\.jdks\openjdk-25.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.2.4\lib\idea_rt.jar=52735" -Dfile.encoding=UTF-8 -
Recorrido INORDEN:
C H R
Recorrido PREORDEN:
R C H
Recorrido POSTORDEN:
H C R
Process finished with exit code 0
```

GitHub: <https://github.com/lizz-mond35/Estructura-de-Datos.git>