

# Documento de diseño - Taller 3

Laura Valentina Ovalle Benitez, Juan Miguel Zuluaga Suárez  
Bogotá, Pontificia Universidad Javeriana

[ovalle.laura@javeriana.edu.co](mailto:ovalle.laura@javeriana.edu.co)  
[jmiguel\\_zuluaga@javeria.edu.co](mailto:jmiguel_zuluaga@javeria.edu.co)

## Diseño de los TADs

### Diseño del árbol AVL :

#### 1. TAD árbol AVL

##### a. Conjunto mínimo de datos

- raíz: Es un apuntador que apunta a la raíz del árbol

##### b. COMPORTAMIENTO

- altura(), calcula la altura de un subárbol que comienza desde el nodo que recibe como parámetro
- diferencia(), calcula la diferencia de altura entre el subárbol derecho e izquierdo de un nodo
- RR\_rotacion(), se encarga de realizar la rotación hacia la derecha en el nodo especificado.
- LL\_rotacion(), se encarga de realizar la rotación hacia la izquierda en el nodo especificado.
- RL\_rotacion(), se encarga de realizar primero una rotación a la derecha y luego a la izquierda.
- LR\_rotacion(), se encarga de realizar primero una rotación a la izquierda y luego a la derecha.
- balance(), se encarga de realizar el balanceo del árbol.
- insertar(), se encarga de insertar un nodo en el árbol y devolver la raíz del árbol actualizada.
- inorder2(), se encarga de ordenar el árbol de manera InOrder y almacenar los datos organizados en un vector.
- Min(), se encarga de encontrar el nodo que contenga el menor valor en el árbol
- Max(), se encarga de encontrar el nodo que contenga el mayor valor en el árbol
- eliminar(), se encarga de eliminar un nodo en específico del árbol
- setRaiz(), fija una nueva raíz para el árbol
- getRaiz(), retorna la raíz actual del árbol
- buscar(), se encarga de buscar un elemento en específico en el árbol, y en caso de encontrarlo retorna el nivel en el que se encuentra dicho elemento
- mediana(), se encarga de calcular la mediana del árbol usando la función buscar.

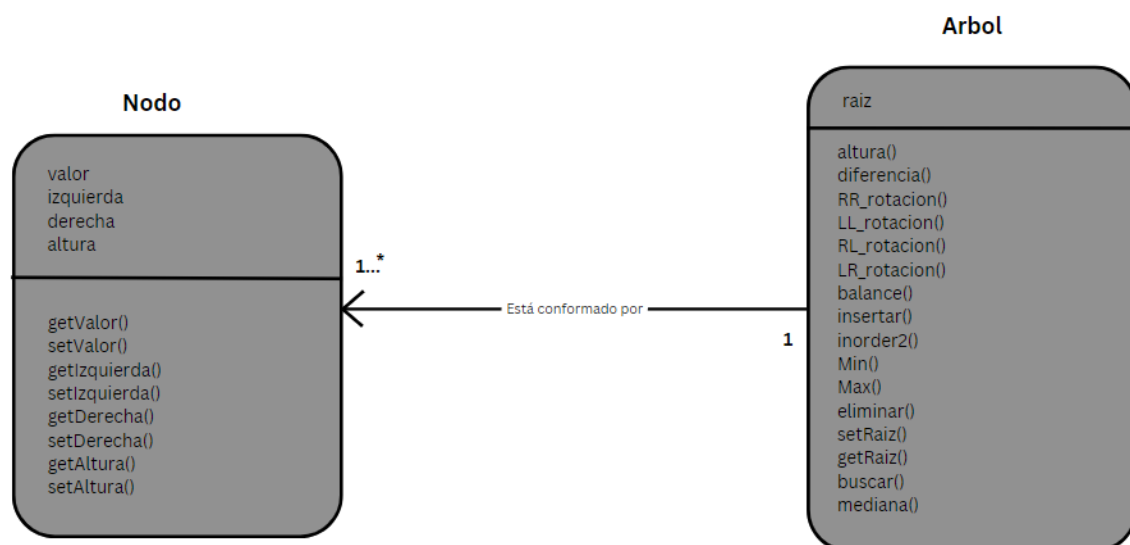
## 2. TAD *Nodo*

### a. CONJUNTO MÍNIMO DE DATOS

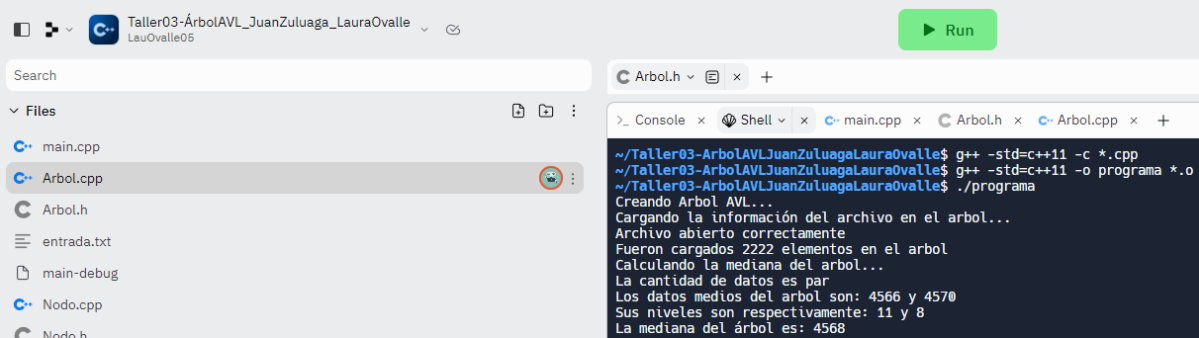
- valor: entero que representa el valor numérico que contiene el nodo
- izquierda: Es un apuntador que siempre apunta al hijo izquierdo de ese nodo
- derecha: Es un apuntador que siempre apunta al hijo derecho de ese nodo
- altura: entero que representa la altura del árbol

### b. COMPORTAMIENTO

- getValor(), retorna el valor que contiene actualmente el nodo
- setValor(), fija un nuevo valor para el nodo
- getIzquierda(), retorna el apuntador al hijo izquierdo del nodo
- setIzquierda(), fija un nuevo apuntador para el hijo izquierdo del nodo
- getDerecha(), retorna el apuntador al hijo derecho del nodo
- setDerecha(), fija un nuevo apuntador para el hijo derecho del nodo
- getAltura(), retorna la altura actual del árbol
- setAltura(), fija un nuevo valor para la altura del árbol



## Prueba funcionamiento del código



The screenshot shows a C++ IDE interface. On the left, a file explorer displays a project named 'Taller03-ArbolAVL\_JuanZuluaga\_LauraOvalle' with files including 'main.cpp', 'Arbol.cpp', 'Arbol.h', 'entrada.txt', 'main-debug', 'Nodo.cpp', and 'Nodo.h'. The 'Arbol.cpp' file is selected. On the right, a terminal window shows the execution of the program. The terminal output indicates that the program was compiled successfully using g++ and executed. The output text is as follows:

```
~/Taller03-ArbolAVL_JuanZuluaga_LauraOvalle$ g++ -std=c++11 -c *.cpp
~/Taller03-ArbolAVL_JuanZuluaga_LauraOvalle$ g++ -std=c++11 -o programa *.o
~/Taller03-ArbolAVL_JuanZuluaga_LauraOvalle$ ./programa
Creando Arbol AVL...
Cargando la información del archivo en el arbol...
Archivo abierto correctamente
Fueron cargados 2222 elementos en el arbol
Calculando la mediana del arbol...
La cantidad de datos es par
Los datos medtos del arbol son: 4566 y 4570
Sus niveles son respectivamente: 11 y 8
La mediana del árbol es: 4568
```