

# DOCUMENTO DE DISEÑO

**Institución:** Pontificia Universidad Javeriana

**Nombre:** Mauricio Rodriguez Cordoba, Tomas De Aza Marquez

## REQUERIMIENTOS PREVIOS:

Archivo .TXT: Un archivo txt que contenga la información de los procesos que se necesitan hacer en el programa con el siguiente formato:

- Instrucción (Agregar O Eliminar): Nos permite saber qué proceso va a realizar el programa.
- Número: Número entero positivo o negativo.

## TADS

### LECTURA

#### DATOS MÍNIMOS:

Acción: String que nos permite guardar la información proveniente del archivo para así mismo ejecutarla en el programa

Valor: Entero que nos permite guardar el valor al que se le ejecutara determinado proceso.

#### OPERACIONES:

Tokenizar: Permite guardar los valores que el archivo .txt provee para poder ejecutarlos en el programa.

### NODO

#### DATOS MÍNIMOS:

- Valor : Número entero ya sea positivo o negativo que contendrá el nodo.
- Node left, right: Apuntador al nodo que nos permitirá movernos por el árbol construido.
- Height: Número entero que mide la altura del árbol.

#### OPERACIONES:

- getHeight(Nodo\*), Retorna la altura del árbol.
- getBalanceFactor(Nodo\*), Retorna el factor de balanceo.
- righthRotate(Nodo\*), Realiza el giro del nodo hacia la derecha.
- leftRotate(Nodo\*), Realiza el giro del nodo hacia la izquierda.
- minValueNode(Nodo\*), Retorna el nodo con el mínimo valor del subárbol deseado.
- deleteNode(Nodo\*, Valor), Elimina el valor deseado del árbol.
- insertNode(Nodo\*, Valor), Insertar el valor deseado en el árbol.
- inorderTraversal(Nodo\*), Imprime los datos recorriendo el árbol en InOrder(LDR).

## DIAGRAMA DE RELACIÓN

LECTURA
<b>ATRIBUTOS:</b> <ul style="list-style-type: none"><li>- Acción</li><li>- Valor</li></ul>
<b>MÉTODOS:</b> <ul style="list-style-type: none"><li>+ Tokenizar (Nombre_Archivo, List &lt;Lectura&gt;)</li></ul>

NODO
<b>ATRIBUTOS:</b> <ul style="list-style-type: none"><li>- Valor</li><li>- Node left, right</li><li>- Height</li></ul>
<b>MÉTODOS:</b> <ul style="list-style-type: none"><li>+ getHeight(Nodo*), Retorna la altura del árbol.</li><li>+ getBalanceFactor(Nodo*), Retorna el factor de balanceo.</li><li>+ righthRotate(Nodo*), Realiza el giro del nodo hacia la derecha.</li><li>+ leftRotate(Nodo*), Realiza el giro del nodo hacia la izquierda.</li><li>+ minValueNode(Nodo*), Retorna el nodo con el mínimo valor del subárbol deseado.</li><li>+ deleteNode(Nodo*, Valor), Elimina el valor deseado del árbol.</li><li>+ insertNode(Nodo*, Valor), Insertar el valor deseado en el árbol.</li><li>+ inorderTraversal(Nodo*), Imprime los datos recorriendo el árbol en InOrder(LDR).</li></ul>