|  |
| --- |
| **TAD Binarie Search Tree <T>** |
| Root={Node= {nodelefth,noderight} ^ root={node1,node2,…..,noden}} |
| {Inv:  Inv: leaves🡪**nodeleft**=**noderight**=**null**} |
| Primitive Operations   * AddNode: Value 🡪Node * SearchNode: NodexT 🡪Boolean * TraverseInOrder: 🡪Text * TraversePreOrder: 🡪Text * DeleteNode: NodexT 🡪 * TraversePosOrden: 🡪Text |

|  |
| --- |
| AddNode(value):  \*Insertst a node into de BST\*  {pre: value!=null}  {post:Node={Value} }+ |

|  |
| --- |
| DeleteNode(value):  \*deletes a node in the BST\*  {pre: true}  {post: node=null; } |

|  |
| --- |
| SearchNode(value):  \*searchs a node into de BST\*  {pre: True}  {post: < node > } |

|  |
| --- |
| TraversePreOrder:  \*Shows the nodes in the tree on a list in pre-order\*  {pre: True}  {post: < text> } |

|  |
| --- |
| TraversePostOrder:  \*Shows the nodes in the tree on a list in post-order\*  {pre: True}  {post: < text > } |

|  |
| --- |
| TraverseInOrder:  \*Shows the nodes in the tree on a list in inorder\*  {pre: True}  {post: < text > ^ noden.valu <noden+1.value } |