**Primer Proyecto Programado.**

Marvin Jesus Calvo Acuña

mcalvoa@ucenfotec.ac.cr

**RESUMEN:***.*

**PALABRAS CLAVE**: Lista, Nodo, Árbol Binario, Árbol binario de búsqueda.

# INTRODUCCIÓN

En este documento se describe los tipos de datos abstractos utilizados en la solución de la tarea de Estructuras de datos 2 impartido en la Universidad Cenfotec. Se presentará de cada estructura de datos creada la representación abstracta,la invariante del TAD y las operaciones del mismo.

# Arbol Binario

## REPRESENTACIÓN ABSTRACTA

<Nodo1, Nodo2, ..., Nodo>,

dónde Curso sea diferente de nulo.

Se puede notar en esta representación como cada objeto Nodo referencia al siguiente y así sucesivamente hasta llegar al nodo que referencia a nada o nulo.

## INVARIANTE

Todos los Nodos son elementos del el árbol binario y contienen nodo derecho, nodo izquierdo y la información que reguarda.

La información no puede estar repetido.

## OPERACIONES

ArbolBinario(Nada)->Nada

(Constructora) //Por defecto devuelve Nada.

**GetInfoRaiz**()->TipoGenerico

(Analizadora)//Retorna un Tipo Generico de Dato.

**InsertarHijoIzquierdo**(T,T)->Boolean

(Analizadora)// Retorna true si logro la inserción del dato de lo contrario false.

**InsertarHijoDerecho**(T,T)->Boolean

(Analizadora)// Retorna true si logro la inserción del dato de lo contrario false.

**Eliminar**(T)->Boolean

(Analizadora)//Retorna true si logro eliminar el dato, de lo contario retorna false.

**EliminarCaso1**(Nodo)->Boolean

(Analizadora)//Retorna true si logro eliminar el dato, de lo contario retorna false.

**EliminarCaso2**(Nodo)->Boolean

(Analizadora)//Retorna true si logro eliminar el dato, de lo contario retorna false.

**EliminarCaso3**(Nodo)->Boolean

(Analizadora)//Retorna true si logro eliminar el dato, de lo contario retorna false.

**MasIzquierdo**(Nodo)->Nodo

(Analizadora)//Retorna el nodo está más a la derecha en el árbol binario.

**GetPadre** (Nodo)->Nodo

(Analizadora)//Retorna el padre del nodo.

**GetPadre** (Nodo,T)->Nodo

(Analizadora)//Retorna el padre del nodo.

**Esta** (Nodo)->Boolean

(Analizadora)//Retorna true si el nodo existe en la estructura de lo contrario retorna false.

**Esta** (Nodo,T)->Boolean

(Analizadora)//Retorna true si el nodo existe en la estructura de lo contrario retorna false.

**InOrden** ()->List

(Analizadora)Retorna una lista con los datos del árbol en forma ordenada.

**InOrden** (Nodo,List)->List

(Analizadora)Retorna una lista con los datos del árbol en forma ordenada.

**PreOrden** ()->List

(Analizadora)Retorna una lista con los datos del árbol en forma pre-ordenada.

**PreOrden** (Nodo,List)->List

(Analizadora)Retorna una lista con los datos del árbol en forma pre-ordenada.

**PosOrden** ()->List

(Analizadora)Retorna una lista con los datos del árbol en forma pos-ordenada.

**PosOrden** (Nodo, List)->List

(Analizadora)Retorna una lista con los datos del árbol en forma pos-ordenada.

## MANEJO DEL ERROR

Todas las operaciones del árbol deben mantener invariante del TAD.

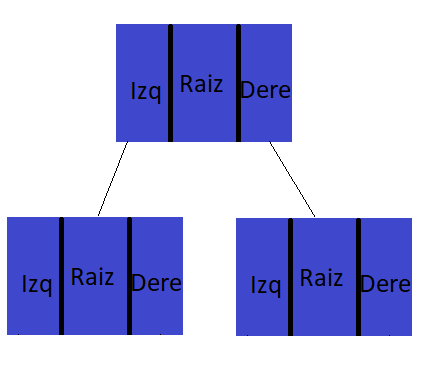
Cuando ocurra una operación indebida en el árbol o que genere error la operación debe abortar su ejecución, y se podría retornar un valor booleano según sea el caso.

# Arbol Binario de Busqueda

## REPRESENTACIÓN ABSTRACTA

<Nodo1, Nodo2, ..., Nodo>,

dónde Curso sea diferente de nulo.



Se puede notar en esta representación como cada objeto Nodo referencia al siguiente y así sucesivamente hasta llegar al nodo que referencia a nada o nulo.

## INVARIANTE

Todos los Nodos son elementos del el árbol binario y contienen nodo derecho, nodo izquierdo y la información que reguarda.

La información no puede estar repetido.

OPERACIONES

**ArbolBinarioDeBusqueda**(Nodo)->Nada(Constructora) //Por defecto nada.

**GetRaiz**( )->Nodo

(Analizadora)//Retorna el nodo Raiz.

**GetRaizInfo**()->T

(Analizadora)//Retorna la información de que contiene el nodo Raiz.

**Insert**(T)->Boolean

(Modificadora)//Retorna True si la inserción se logró de lo contrario retorna falso.

**Insert**(Nodo, T)->Nodo

(Modificadora)//Retorna el nodo insertado.

**Eliminar**(T)->Boolean

(Modificadora)// Elimina un nodo del árbol, de lo contrario retorna faso.

**EliminarArbolBinario**(Nodo,T)->Nada (Modificadora)// Elimina un nodo del árbol y retorna el nodo eliminado.

**EstaArbolBinario**(T)->Boolena

(Analizadora)//Busca el dato y si lo encuentra retorna el true de lo contrario retorna false.

**Buscar**(T)->Nodo

(Analizadora)//Busca el dato y si lo encuentra retorna el nodo de lo contrario retorna null.

**Buscar**(Nodo,T)->Nodo

(Analizadora)//Busca el dato y si lo encuentra retorna el nodo de lo contrario retorna null.

**InOrden** ()->List

(Analizadora)Retorna una lista con los datos del árbol en forma ordenada.

**PreOrden** ()->List

(Analizadora)Retorna una lista con los datos del árbol en forma pre-ordenada.

**PosOrden** ()->List

(Analizadora)Retorna una lista con los datos del árbol en forma pos-ordenada.

**ImprimirArbolBinario** ()->Nada

(Analizadora)Retorna una lista con los datos del árbol en forma pos-ordenada.

## MANEJO DEL ERROR

Todas las operaciones del árbol deben mantener invariante del TAD.

Cuando ocurra una operación indebida en el árbol o que genere error la operación debe abortar su ejecución, y se podría retornar un valor booleano según sea el caso.

# REFERENCIAS

[1] Profe.Maria Eugenia Ucrós,Profe.Christian Sibaja Fernández ,UniversidadCenfotec.Presentacion .Estructuras de Datos TDS1402 y TDS2311.