



Unidad 1

lenguajes y automatas 2

Nombre:Christiam Ivan Rodriguez Moreno

numero de control 13 480619

Catedratico:Juan pablo Rosas Baldazo

# Introduccion

El Programa crea un arbol binario lo que hace es crear nodos y empieza de arriba hacia abajo y ayuda a facilitar la obtencion de ecuaciones y te ayuda a obtener las

*¿Que Hicimos?*

## • Descripcion •

Se busco un codigo que nos ayudara a la implementacion de un arbol binario se encontro un codigo fuente que era de Luis García tomamos la referencia del codigo y nos pusimos a checar como funcionaba el codigo a continuacion sera una muestra que estara implementado en un pseudocodigo

## ¿COMO LO HICIMOS?

Buscamos un codigo que nos ayudara por medio de la web e investigamos que hacia dicho codigo y si era apto para lo que necesitabamos hacer despues de eso empezamos a crear el reporte y convertimos el codigo en pseudocodigo



```
class node():
    se define la clase( dato):
    se ingresa el nodo left
    se ingresa el nodoright
    se declara la variable dato
class arbol():
    define nit(self):
    define self.root
    def insert(self, a, dato):
    if a == None:
    a = node(dato)
    else:
    d = a.dato
    if dato < d:
    a.left = self.insert(a.left, dato)
    else:
    a.right = self.insert(a.right, dato)
    return a
    def inorder(self, a):
    if a == None:
    return None
    else:
    self.inorder(a.left)
    print(a.dato)
    self.inorder(a.right)
    def preorder(self, a):
    if a == None:
    return None
    else:
    print(a.dato)
    self.preorder(a.left)
    self.preorder(a.right)
    def postorder(self, a):
    if a == None:
    return None
    else:
    self.postorder(a.left)
    self.postorder(a.right)
    print(a.dato)
    def buscar(self, dato, a):
    if a == None:
    return None
    else:
```

```

if dato == a.dato:
    return a.dato
else:
    if dato < a.dato:
        return self.buscar(dato, a.left)
    else:
        return self.buscar(dato, a.right)
tree = arbol()
while True:
    os.system("cls")
    print("Arbol ABB")
    opc = input("\n1.-Insertar nodo \n2.-Inorden \n3.-Preorden \n4.-Postorden \n5.-Buscar \n6.-Salir\n\nElige una opcion -> ")
    if opc == '1':
        nodo = input("\nIngresa el nodo -> ")
        if nodo.isdigit():
            nodo = int(nodo)
            tree.root = tree.insert(tree.root, nodo)
        else:
            print("\nIngresa solo digitos...")
    elif opc == '2':
        if tree.root == None:
            print("Vacio")
        else:
            tree.inorder(tree.root)
    elif opc == '3':
        if tree.root == None:
            print("Vacio")
        else:
            tree.preorder(tree.root)
    elif opc == '4':
        if tree.root == None:
            print("Vacio")
        else:
            tree.postorder(tree.root)
    elif opc == '5':
        nodo = input("\nIngresa el nodo a buscar -> ")
        if nodo.isdigit():
            nodo = int(nodo)
            if tree.buscar(nodo, tree.root) == None:
                print("\nNodo no encontrado...")
            else:
                print("\nNodo encontrado -> ", tree.buscar(nodo, tree.root), " si existe...")
        else:

```

```
print("\nIngresa solo digitos...")
elif opc == '6':
print("\nElegiste salir...\n")
os.system("pause")
break
else:
print("\nElige una opcion correcta...")
print()
os.system("pause")
```

#### Conclusion

llegamos a la conclusion que se pueden agregar nodos siempre y cuando los numeros sean enteros y sea por logica

#### Referencias

<https://gist.github.com/codigosdeprogra/12be086b79730718d8fe530d300983b7>