

Instituto Tecnológico de Culiacán



**TECNOLÓGICO
NACIONAL DE MÉXICO**



Carrera: Ingeniería en Sistemas Computacionales

Materia: Inteligencia Artificial

Profesor: Zuriel Dathan Mora Felix

**Tarea: Completar el proyecto: Árbol Binario de Búsqueda
Explicación de Dos Pruebas de Funcionamiento**

Grupo:

11:00 AM – 12:00 PM

Equipo :

García Pérez José Ángel

Verdugo Bermúdez Sebastián

Explicación de Funcionamiento

Inserción de datos al árbol binario

```
class Arbol:
    def __init__(self, raiz=None):
        if raiz is not None:
            self.raiz = Nodo(raiz) # Si recibe un valor, crea un nodo raíz
        else:
            self.raiz = None # Si no, deja el árbol vacío

# INSERTAR DATOS EN EL ARBOL

def insertar_recursivo(self, nodo, dato):
    if dato < nodo.dato:
        if nodo.izquierda is None:
            nodo.izquierda = Nodo(dato)
        else:
            self.insertar_recursivo(nodo.izquierda, dato)
    else:
        if nodo.derecha is None:
            nodo.derecha = Nodo(dato)
        else:
            self.insertar_recursivo(nodo.derecha, dato)
```

```
Inteligencia Artificial > mainarbol.py > ...
1  from arbol import Arbol
2  arbol_numeros = Arbol()
3  arbol_numeros.agregar(1984)
4  arbol_numeros.agregar(60)
5  arbol_numeros.agregar(10)
6  arbol_numeros.agregar(20)
7  arbol_numeros.agregar(10)
8  arbol_numeros.agregar(25)
9  arbol_numeros.agregar(59)
10 arbol_numeros.agregar(64)
11 arbol_numeros.agregar(10)
12 arbol_numeros.agregar(19)
13 arbol_numeros.agregar(23)
14 arbol_numeros.agregar(18)
15 arbol_numeros.agregar(1)
16 arbol_numeros.agregar(2013)
17 arbol_numeros.inorden()
18 arbol_numeros.preorden()
19 arbol_numeros.postorden()
```

Imprimiendo árbol preorden.

1984, 60, 10, 1, 20, 10, 10, 19, 18, 25, 23, 59, 64, 2013,

La función comienza declarando la palabra `init` que sirve para instanciar un dato en este caso será nuestro dato raíz, pero también se valida que puede estar vacío. Una vez inicializado esta función creamos la que insertara los datos en cada nodo ya sea al izquierdo o al derecho del árbol, es una función recursiva por lo que se busca a si misma para el ordenamiento. Comienza preguntando si el dato raíz es menor que dato del nodo izquierdo si está vacío entonces se agrega a ese nodo y sería el primero de la izquierda si en dado caso no está vacío entonces se llama a la misma función recursivamente para intentar insertar

el dato en el subárbol izquierdo (es decir, se repite el proceso con el nodo izquierdo como nuevo nodo actual). Lo mismo sucede para el lado derecho.

Imprimir datos al árbol binario (inorden, preorden y postorden)

```
# IMPRIMIR EL ARBOL EN ORDEN
def inorden_recursivo(self, nodo):
    if nodo is not None:
        self.inorden_recursivo(nodo.izquierda)
        print(nodo.dato, end=", ")
        self.inorden_recursivo(nodo.derecha)

# IMPRIMIR EL ARBOL EN PREORDEN
def preorden_recursivo(self, nodo):
    if nodo is not None:
        print(nodo.dato, end=", ")
        self.preorden_recursivo(nodo.izquierda)
        self.preorden_recursivo(nodo.derecha)

# IMPRIMIR EL ARBOL EN POSTORDEN
def postorden_recursivo(self, nodo):
    if nodo is not None:
        self.postorden_recursivo(nodo.izquierda)
        self.postorden_recursivo(nodo.derecha)
        print(nodo.dato, end=", ")
```

```
def inorden(self):
    print("Imprimiendo árbol inorden: ")
    self.inorden_recursivo(self.raiz)
    print("")

def preorden(self):
    print("Imprimiendo árbol preorden: ")
    self.preorden_recursivo(self.raiz)
    print("")

def postorden(self):
    print("Imprimiendo árbol postorden: ")
    self.postorden_recursivo(self.raiz)
    print("")
```

```
Imprimiendo árbol inorden:
1, 10, 10, 10, 18, 19, 20, 23, 25, 59, 60, 64, 1984, 2013
Imprimiendo árbol preorden:
1984, 60, 10, 1, 20, 10, 10, 19, 18, 25, 23, 59, 64, 2013
Imprimiendo árbol postorden:
1, 18, 19, 10, 10, 23, 59, 25, 20, 10, 64, 60, 2013, 1984
```

En esta parte del código realizamos la impresión del árbol usando métodos por ejemplo en el se hace en inorden es decir ordenado como debe ser primero pregunta si el nodo no es None se llama al propio método ya que es recursivo, pero del nodo de la parte izquierda es decir lo visita el subárbol izquierdo, después imprime el nodo actual y seguido visita el subárbol derecho del mismo modo que el anterior mencionado. En el preorden cambia la forma primero se imprime el nodo actual después el subárbol izquierdo y al último el subárbol derecho, en el postorden se recorre el subárbol izquierdo después el subárbol derecho y finalmente el nodo actual.