



**UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO**

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

INGENIERÍA EN COMPUTACIÓN

Estructura de Datos

Profesor. Jesús Hernández Cabrera

Grupo: 1360

Castro Vázquez Luis Alfredo

```
File Edit Selection View Go Run Terminal Help
NodoArbol.py X ArbolMain.py
Tarea-10 > NodoArbol.py > NodoArbol > set_hijo_izq
1 class NodoArbol:
2     def __init__(self, dato=None, hijo_izq=None, hijo_der=None):
3         self.dato = dato
4         self.hijo_izq = hijo_izq
5         self.hijo_der = hijo_der
6
7     def get_dato(self):
8         return self.dato
9
10    def set_dato(self, dato):
11        self.dato = dato
12
13    def get_hijo_izq(self):
14        return self.hijo_izq
15
16    def set_hijo_izq(self, hijo_izq):
17        self.hijo_izq = hijo_izq
18
19    def get_hijo_der(self):
20        return self.hijo_der
21
22    def set_hijo_der(self, hijo_derecho):
23        self.hijo_der = hijo_derecho
24
25    def __eq__(self, other):
26        if isinstance(other, NodoArbol):
27            return self.dato == other.dato
28        return False
29
30    def __hash__(self):
31        return hash(self.dato)
32
33    def __str__(self):
34        return f"NodoArbol(dato={self.dato}, hijo_izquierdo={self.hijo_izq}, hijo_derecho={self.hijo_der})"
35
Ln 16, Col 38 Spaces: 4 UTF-8 CRLF Python 3.12.3 Prettier
```

```
File Edit Selection View Go Run Terminal Help
NodoArbol.py ArbolMain.py M
Tarea-10 > ArbolMain.py > ArbolMain
1 from NodoArbol import NodoArbol
2
3 class ArbolMain:
4     def arbol_numeros(self):
5         nodo1 = NodoArbol(1)
6         nodo2 = NodoArbol(5, hijo_izq=nodo1)
7         nodo3 = NodoArbol(25)
8         nodo4 = NodoArbol(15, hijo_der=nodo3)
9         raiz = NodoArbol(10, hijo_izq=nodo2, hijo_der=nodo4)
10        return raiz
11
12    def arbol_nombres(self):
13        nodo1 = NodoArbol("Susan")
14        nodo2 = NodoArbol("Diana")
15        nodo3 = NodoArbol(
16            "Pedro", hijo_izq=nodo1, hijo_der=nodo2)
17        nodo4 = NodoArbol("Mario")
18        raiz = NodoArbol(
19            "Diego", hijo_izq=nodo3, hijo_der=nodo4)
20        return raiz
21
22    def main():
23        arbol_main = ArbolMain()
24
25        arbol_numeros = arbol_main.arbol_numeros()
26        print("Arbol de números:")
27        print(arbol_numeros)
28
29        arbol_nombres = arbol_main.arbol_nombres()
30        print("\nArbol de nombres:")
31        print(arbol_nombres)
32
33
34
35 if __name__ == "__main__":
36     main()
37
Ln 11, Col 1 Spaces: 4 UTF-8 CRLF Python 3.12.3 Prettier
```

Herramienta Recortes

Captura de pantalla copiada en el portapapeles
Guardado automáticamente en la carpeta de
capturas de pantalla.

Markup and share

The image shows a Visual Studio Code editor window with a Python file named `ArbolMain.py` open. The file contains a class `ArbolMain` with a method `def arbol_numeros(self):` that constructs a binary tree of numbers. The terminal output shows the execution of the script, which prints the tree structure for numbers and names.

```
1 from NodoArbol import NodoArbol
2
3
4
5 class ArbolMain:
6     def arbol_numeros(self):
7         nodo1 = NodoArbol(1)
8         nodo2 = NodoArbol(5, hijo_izq=nodo1)
9         nodo3 = NodoArbol(25)
```

Terminal Output:

```
PS C:\Users\luisa\OneDrive\Escritorio\EDD_2025> & C:/Users/luisa/AppData/Local/Programs/Python/Python312/python.exe c:/Users/luisa/OneDrive/Escritorio/EDD_2025/Tarea-10/ArbolMain.py
Árbol de números:
NodoArbol(dato=10, hijo_izquierdo=NodoArbol(dato=5, hijo_izquierdo=NodoArbol(dato=1, hijo_izquierdo=None, hijo_derecho=None), hijo_derecho=NodoArbol(dato=15, hijo_izquierdo=None,
hijo_derecho=NodoArbol(dato=25, hijo_izquierdo=None, hijo_derecho=None)))

Árbol de nombres:
NodoArbol(dato=Diego, hijo_izquierdo=NodoArbol(dato=Pedro, hijo_izquierdo=NodoArbol(dato=Susan, hijo_izquierdo=None, hijo_derecho=None), hijo_derecho=NodoArbol(dato=Diana, hijo_izquierdo=None,
hijo_derecho=None)), hijo_derecho=NodoArbol(dato=Mario, hijo_izquierdo=None, hijo_derecho=None))
PS C:\Users\luisa\OneDrive\Escritorio\EDD_2025>
```