

Tarea 2

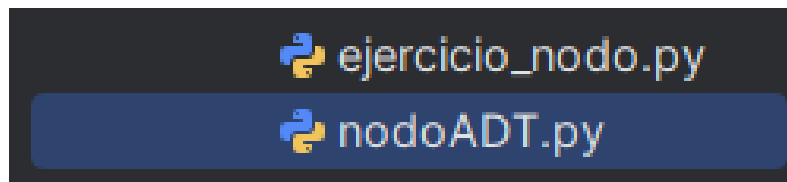
NODO ADT

Estructura de Datos
Universidad Nacional Autónoma de México
Facultad de Estudios Superiores Aragón

REALIZADO POR:

**Enrique Emiliano Cano
García**

Se divide en 2 archivos para una mejor visualización (nodoADT.py ,
ejercicio_nodo.py)



Se definen la clase y con la características de Python el constructor se genera sobre la clase

```
1 class NodoADT: 8 usages  👤 kno4
2     def __init__(self, dato = None, siguiente = None):  👤 kno4
3         self.dato = dato
4         self.siguiente = siguiente
5
6     def __str__(self):  👤 kno4
7         return f'{self.dato} {self.siguiente}'
8
1 from Tareas.Tarea3.nodoADT import NodoADT
2 def main(): 1 usage  👤 kno4
3     head_5 = NodoADT(600)
4     head_4 = NodoADT(400, head_5)
5     head_3 = NodoADT(300, head_4)
6     head_2 = NodoADT(200, head_3)
7     head = NodoADT(100, head_2)
8     print(head)
9     print("-----")
10    head_3.dato = 333
11    print(head)
12    print("-----")
13    head_6 =NodoADT(700)
14    head_5.siguiente = head_6
15    print(head)
16    print("-----")
17    head_0 = NodoADT(50, head)
18    print(head_0)
19    if __name__ == '__main__':
20        main()
21
```

Sobre la consola y con los prints hechos podemos ver exactamente lo que se requiere aunque no sea exactamente como se indicó

```
100 200 300 400 600 None
-----
100 200 333 400 600 None
-----
100 200 333 400 600 700 None
-----
50 100 200 333 400 600 700 None

Process finished with exit code 0
```