

```
class NodoDoble:
    def __init__(self, dato, anterior = None, siguiente = None):
        self.dato = dato
        self.anterior = anterior
        self.siguiente = siguiente

    def get_dato(self):
        return self.dato

    def get_siguiente(self):
        return self.siguiente

    def get_anterior(self):
        return self.anterior

    def set_anterior(self, anterior):
        self.anterior = anterior

    def set_siguiente(self, siguiente):
        self.siguiente = siguiente

    def set_dato(self, dato):
        self.dato = dato
```

```
class DoubleLinkedList:
    def __init__(self):
        self.cabeza = None
        self cola = None

    def esta_vacia(self):
        if self.cabeza is None:
            vacia = True
        else:
            vacia = False
        return vacia

    def get_tamano(self):
        contador = 0
        actual = self.cabeza
        while actual is not None:
            contador += 1
            actual = actual.siguiente
        return contador
```

```

def agregar_al_inicio(self, dato):

    nuevo_nodo = NodoDoble(dato)
    if self.cabeza is None:
        self.cabeza = self cola = nuevo_nodo
    else:
        nuevo_nodo.siguiente = self.cabeza
        self.cabeza = nuevo_nodo

def agregar_al_final(self, dato):
    nuevo_nodo = NodoDoble(dato)
    if self.cabeza is None:
        self.cabeza = self.col = nuevo_nodo
    else:
        self.col.siguiente = nuevo_nodo
        self.col = nuevo_nodo

def agregar_despues_de(self, referencia, dato):
    nuevo_nodo = NodoDoble(dato)
    actual = self.cabeza
    while actual.dato != referencia:
        actual = actual.siguiente
    nuevo_nodo.siguiente = actual.siguiente
    actual.siguiente = nuevo_nodo

def obtener_posicion(self, posicion):
    contador = 2
    actual = self.cabeza
    if posicion > 1:
        while contador < posicion + 1:
            actual = actual.siguiente
            contador += 1
    print(f'En la posicion {posicion} se encuentra {actual.get_dato()}')

def eliminar_el_primero(self):
    actual = self.cabeza
    self.cabeza = actual.get_siguiente()

def eliminar_el_final(self):
    self.col = self.col.anterior
    self.col.siguiente = None

def eliminar_posicion(self, posicion):
    actual = self.cabeza

```

```

contador = 1
if posicion == 1:
    self.cabeza = actual.get_siguiete()
else:
    while contador < posicion - 1:
        actual = actual.get_siguiete()
        contador += 1
    actual.set_siguiete(actual.get_siguiete().get_siguiete())

```

```

def buscar_valor(self, dato):
    actual = self.cabeza
    contador = 1
    while actual.dato != dato:
        actual = actual.get_siguiete()
        contador += 1
    print(f'{dato} se encuentra en la posicion {contador}')

```

```

def actualizar(self,a_buscar,dato):
    contador = 1
    actual = self.cabeza
    if a_buscar > 1:
        while contador < a_buscar:
            actual = actual.get_siguiete()
            contador += 1
    actual.set_dato(dato)

```

```

def transversal(self):
    actual = self.cabeza
    while actual is not None:
        print(actual.dato, end=' --> ')
        actual = actual.get_siguiete()
    print('None')

```

---

```

from tarea5 import *

```

```

valores = DoubleLinkedList()

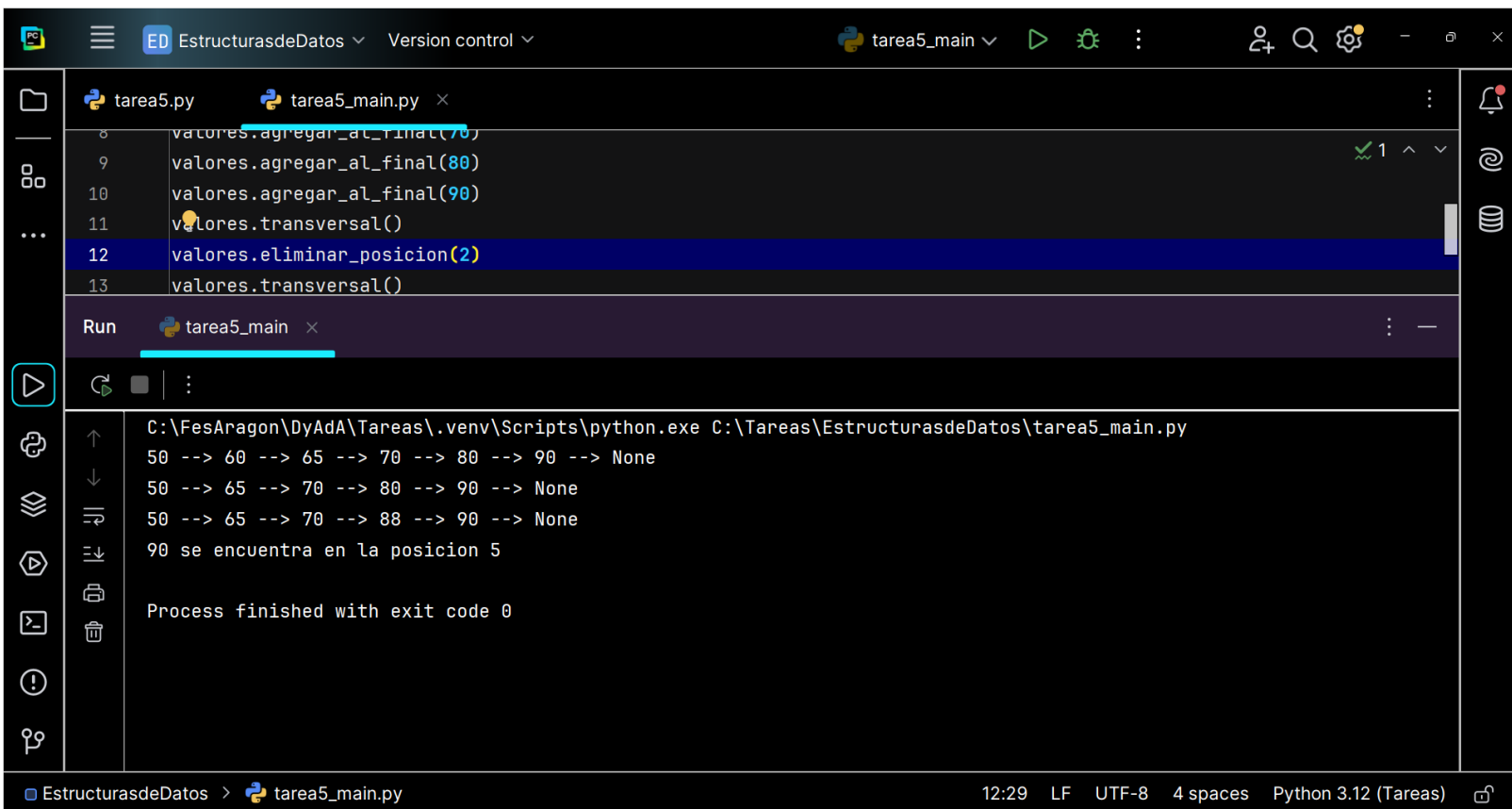
```

```

valores.agregar_al_inicio(50)
valores.agregar_al_final(60)
valores.agregar_al_final(65)
valores.agregar_al_final(70)
valores.agregar_al_final(80)
valores.agregar_al_final(90)
valores.transversal()
valores.eliminar_posicion(2)

```

```
valores.transversal()
valores.actualizar(4, 88)
valores.transversal()
valores.buscar_valor(90)
```



The screenshot shows a Visual Studio Code editor window with the following components:

- Editor Tabs:** Two tabs are open: `tarea5.py` and `tarea5_main.py`. The `tarea5_main.py` tab is active.
- Code Editor:** The active file contains the following Python code:

```
8 valores.agregar_al_final(70)
9 valores.agregar_al_final(80)
10 valores.agregar_al_final(90)
11 valores.transversal()
12 valores.eliminar_posicion(2)
13 valores.transversal()
```

Line 12 is selected.
- Run and Debug Console:** The console shows the execution of `tarea5_main.py`. The output is:

```
C:\FesAragon\DyAdA\Tareas\.venv\Scripts\python.exe C:\Tareas\EstructurasdeDatos\tarea5_main.py
50 --> 60 --> 65 --> 70 --> 80 --> 90 --> None
50 --> 65 --> 70 --> 80 --> 90 --> None
50 --> 65 --> 70 --> 88 --> 90 --> None
90 se encuentra en la posicion 5

Process finished with exit code 0
```
- Bottom Status Bar:** Shows the current file path `EstructurasdeDatos > tarea5_main.py`, the time `12:29`, the line ending `LF`, the encoding `UTF-8`, the indentation `4 spaces`, and the Python version `Python 3.12 (Tareas)`.