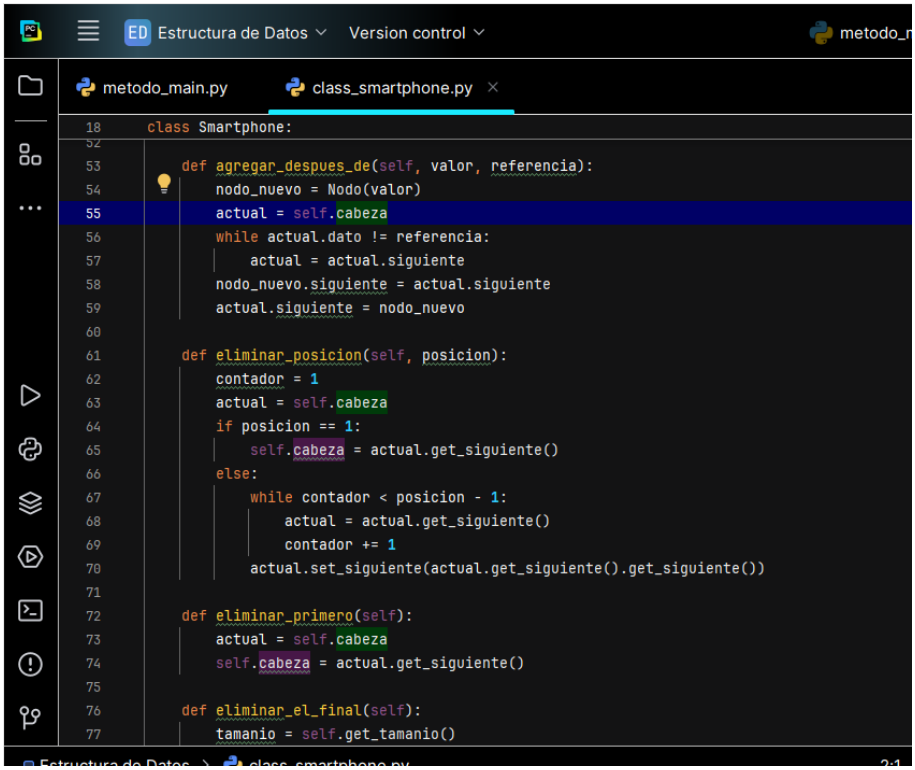
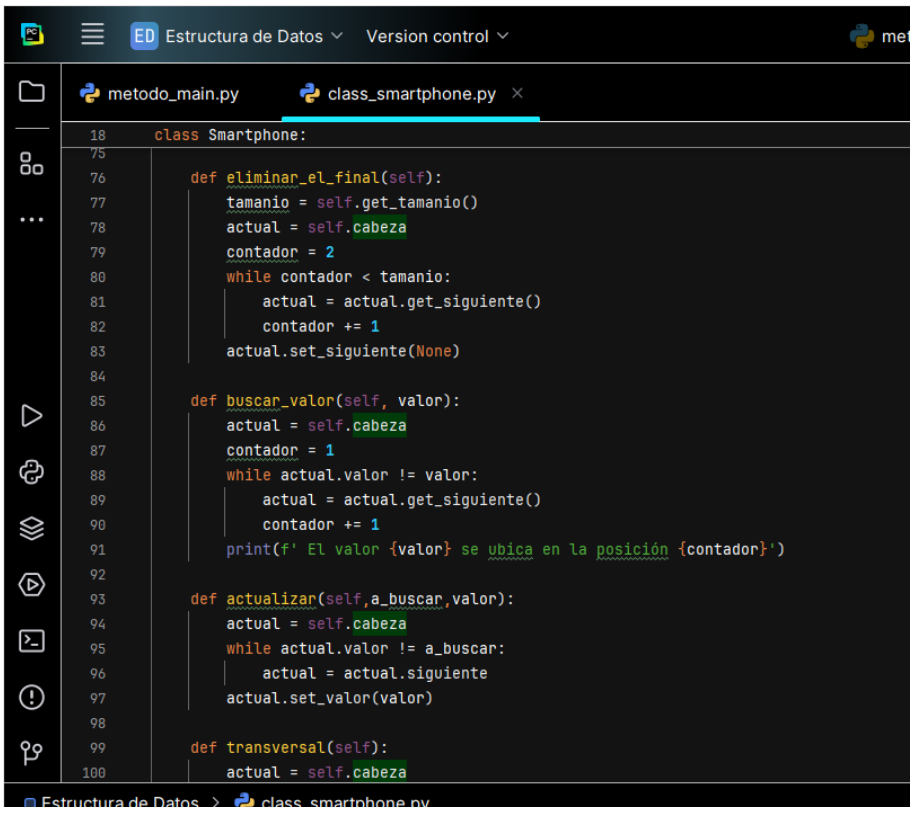
metodo_main.py class_smartphone.py

```python
class Nodo:
    def __init__(self, valor, siguiente= None):
        self.valor = valor
        self.siguiente = siguiente

    def get_valor(self):
        return self.valor

    def set_valor(self, valor):
        self.valor = valor

    def get_siguiente(self):
        return self.siguiente

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

class Smartphone:
    def __init__(self):
        self.cabeza = None

    def esta_vacia(self):
        if self.cabeza is None:
            print('Está vacia')
        else:
            print('No está vacia')

```

metodo_main.py class_smartphone.py

```
18  class Smartphone:
27
28      def get_tamanio(self):
29          contador = 0
30          actual = self.cabeza
31          while actual is not None:
32              contador += 1
33              actual = actual.get_siguiente()
34          return contador
35
36      def agregar_al_final(self, dato):
37          nodo_nuevo = Nodo(dato)
38          if self.cabeza is None:
39              self.cabeza = nodo_nuevo
40          else:
41              actual = self.cabeza
42              while actual.get_siguiente() is not None:
43                  actual = actual.siguiente
44              actual.set_siguiente(nodo_nuevo)
45
46      def agregar_al_inicio(self, valor):
47          nodo_nuevo = Nodo(valor)
48          nodo_nuevo.siguiente = self.cabeza
49          self.cabeza = nodo_nuevo
50
51
52
```

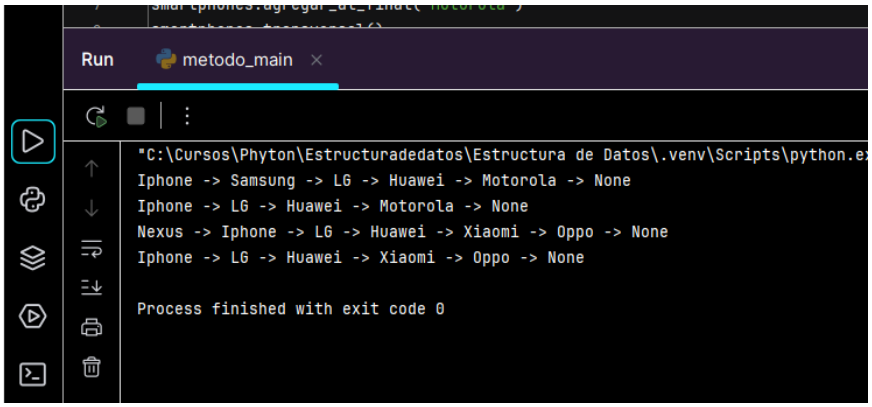

Estructura de Datos
Version control
metodo_main.py
class_smartphone.py
class Smartphone:
def agregar_despues_de(self, valor, referencia):
nodo_nuevo = Nodo(valor)
actual = self.cabeza
while actual.dato != referencia:
actual = actual.siguiente
nodo_nuevo.siguiente = actual.siguiente
actual.siguiente = nodo_nuevo
def eliminar_posicion(self, posicion):
contador = 1
actual = self.cabeza
if posicion == 1:
self.cabeza = actual.get_siguiente()
else:
while contador < posicion - 1:
actual = actual.get_siguiente()
contador += 1
actual.set_siguiente(actual.get_siguiente().get_siguiente())
def eliminar_primero(self):
actual = self.cabeza
self.cabeza = actual.get_siguiente()
def eliminar_el_final(self):
tamanio = self.get_tamanio()


Estructura de Datos
Version control
metodo_main.py
class_smartphone.py
class Smartphone:
def eliminar_el_final(self):
tamanio = self.get_tamanio()
actual = self.cabeza
contador = 2
while contador < tamanio:
actual = actual.get_siguiente()
contador += 1
actual.set_siguiente(None)
def buscar_valor(self, valor):
actual = self.cabeza
contador = 1
while actual.valor != valor:
actual = actual.get_siguiente()
contador += 1
print(f' El valor {valor} se ubica en la posición {contador}')
def actualizar(self,a_buscar,valor):
actual = self.cabeza
while actual.valor != a_buscar:
actual = actual.siguiente
actual.set_valor(valor)
def transversal(self):
actual = self.cabeza

```python
class Smartphone:

    def buscar_valor(self, valor):
        actual = self.cabeza
        contador = 1
        while actual.valor != valor:
            actual = actual.get_siguiente()
            contador += 1
        print(f' El valor {valor} se ubica en la posición {contador}')

    def actualizar(self,a_buscar,valor):
        actual = self.cabeza
        while actual.valor != a_buscar:
            actual = actual.siguiente
        actual.set_valor(valor)

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

Método main

```python
from class_smartphone import *
smartphones = Smartphone()
smartphones.agregar_al_final('Iphone')
smartphones.agregar_al_final('Samsung')
smartphones.agregar_al_final('LG')
smartphones.agregar_al_final('Huawei')
smartphones.agregar_al_final('Motorola')
smartphones.transversal()
smartphones.eliminar_posicion(2)
smartphones.transversal()
smartphones.actualizar(a_buscar: 'Motorola', valor: 'Xiaomi')
smartphones.agregar_al_inicio('Nexus')
smartphones.agregar_al_final('Oppo')
smartphones.transversal()
smartphones.eliminar_primero()
smartphones.transversal()
```


Run
metodo_main
"C:\Cursos\Phyton\Estructuradedatos\Estructura de Datos\.venv\Scripts\python.ex
Iphone -> Samsung -> LG -> Huawei -> Motorola -> None
Iphone -> LG -> Huawei -> Motorola -> None
Nexus -> Iphone -> LG -> Huawei -> Xiaomi -> Oppo -> None
Iphone -> LG -> Huawei -> Xiaomi -> Oppo -> None
Process finished with exit code 0