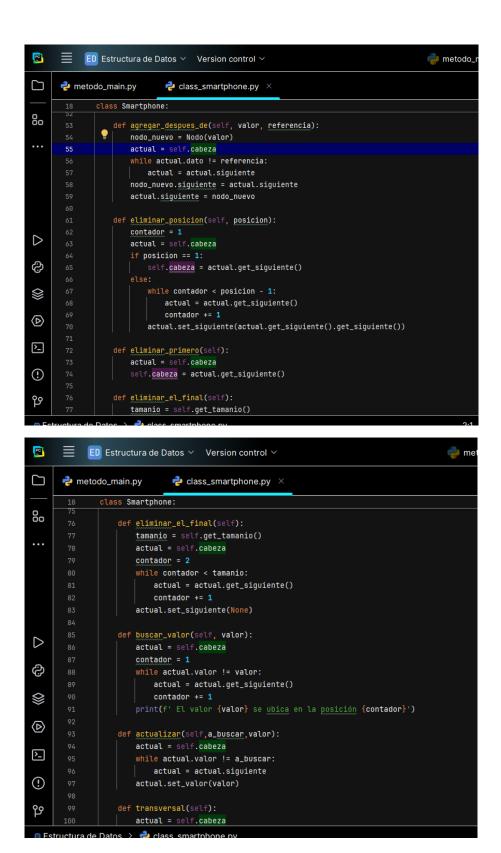
```
netodo_main.py
                       class_smartphone.py ×
       class Nodo:
           def __init__(self, valor, siguiente= None):
               self.valor = valor
               self.siguiente = siguiente
           def get_valor(self):
               return self.valor
           def set_valor(self, valor):
               self.valor = valor
           def get_siguiente(self):
               return self.siguiente
           def set_siguiente(self, siguiente):
               self.siguiente = siguiente
       class Smartphone:
               self.cabeza = None
           def esta_vacia(self):
               if self.cabeza is None:
                  print('No está vacia')
```

```
class_smartphone.py ×
metodo_main.py
       class Smartphone:
28
           def get_tamanio(self):
               contador = 0
30
               actual = self.cabeza
               while actual is not None:
                   contador += 1
                   actual = actual.get_siguiente()
               return contador
35
           def agregar_al_final(self, dato):
               nodo_nuevo = Nodo(dato)
               if self.cabeza is None:
                  self.cabeza = nodo_nuevo
                   actual = self.cabeza
                   while actual.get_siguiente() is not None:
                       actual = actual.siguiente
                   actual.set_siguiente(nodo_nuevo)
           def agregar_al_inicio(self, valor):
               nodo_nuevo = Nodo(valor)
               nodo_nuevo.siguiente = self.cabeza
               self.cabeza = nodo_nuevo
```



```
class Smartphone:
00
                    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
\Diamond
                        while actual.valor != a_buscar:
                            actual = actual.siguiente
9
                        actual.set_valor(valor)
                   def transversal(self):
\otimes
                        actual = self.cabeza
                        while actual is not None:
(4)
                            print(actual.valor, end=' -> ')
                            actual = actual.get_siguiente()
\overline{\mathsf{N}}
                        print('None')
```

Método main

```
RC.
           ED Estructura de Datos Version control V
netodo_main.py × dess_smartphone.py
              from class_smartphone import *
00
             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')
\Diamond
             smartphones.agregar_al_final('Oppo')
             smartphones.transversal()
9
             smartphones.eliminar_primero()
             smartphones.transversal()
\otimes
```

