



*Guevara Paniagua David.*



*Grupo 1360.*

*Profesor Hernández Cabrera Jesús.*

*Estructura de datos.*

*Tarea 4: Lista Ligada.*

Creación de la clase *LinkedList*:

```
from Nodo import *

class LinkedList:
    def __init__(self):
        self.head=None
        self.tamanio=0
    def esta_vacia(self):
        flag=False
        if self.tamanio==0:
            flag = True
        return flag
    def get_tamanio(self):
        return self.tamanio
```

```
def agregar_al_final(self,dato):
    if self.esta_vacia():
        self.head=Nodo(dato,None)
    else:
        aux=self.head
        while aux.getSiguiente()!=None:
            aux=aux.getSiguiente()
        aux.setSiguiente(Nodo(dato,None))
    self.tamanio+=1
def agregar_al_inicio(self,dato):
    if self.esta_vacia():
        self.head=Nodo(dato,None)
    else:
        self.head=Nodo(dato,self.head)
    self.tamanio+=1
def agregar_despues_de(self,referencia,dato):
    posicionDeReferencia=self.buscar(referencia)
    if posicionDeReferencia != None:
        aux=self.head
        for i in range(posicionDeReferencia):
            aux=aux.getSiguiente()
        aux.setSiguiente(Nodo(dato,aux.getSiguiente()))
    self.tamanio+=1
```

```

def eliminar(self, posicion):
    if posicion < self.get_tamano() and posicion >= 0:
        if posicion==0:
            self.head=self.head.getSiguiente()
        else:
            aux=self.head
            for i in range(posicion-1):
                aux=aux.getSiguiente()
            aux.setSiguiente(aux.getSiguiente().getSiguiente())
            self.tamano-=1
    else:
        print("Posicion fuera del rango!")
def eliminar_el_primer(self):
    self.head=self.head.getSiguiente()
    self.tamano-=1
def eliminar_el_final(self):
    aux=self.head
    for i in range(self.tamano-2):#se descuenta el head y el ultimo para llegar al penultimo
        aux=aux.getSiguiente()
    aux.setSiguiente(None)
    self.tamano-=1

```

```

def buscar(self, dato):
    aux=self.head
    currentPos=0
    while aux!=None and aux.getValor()!=dato:
        currentPos+=1
        aux=aux.getSiguiente()
    if aux==None:
        currentPos=None
        print("El valor", dato, "no fue encontrado!!")
    return currentPos
def actualizar(self, a_buscar, dato):
    a_buscarPosicion=self.buscar(a_buscar)
    if a_buscarPosicion!=None:
        aux=self.head
        for i in range(a_buscarPosicion):
            aux=aux.getSiguiente()
        aux.setValor(dato)

def transversal(self):
    aux=self.head
    while aux!=None:
        print(aux.getValor().toString(), "--> ", end="")
        aux=aux.getSiguiente()
    print("None")

```

Creación de la clase *Smartphone*:

```
class Smartphone:
    def __init__(self,marca,capacidad):
        self.marca=marca
        self.capacidad=capacidad
    def toString(self):
        cadena="|Marca: "+self.marca+", Capacidad: "+str(self.capacidad)+"GB|"
        return cadena
    def getMarca(self):
        return self.marca
    def setMarca(self,marca):
        self.marca=marca
    def getCapacidad(self):
        return self.capacidad()
    def setCapacidad(self,capacidad):
        self.capacidad=capacidad
```

---

Archivo *main* con las acciones a ejecutar:

```
from Smartphone import *
from Nodo import *
from LinkedList import *
#Creando 5 smartphone
tel1=Smartphone("Nokia",120)
tel2=Smartphone("Xiamoi",256)
tel3=Smartphone("Samsung",500)
tel4=Smartphone("Motorola",80)
tel5=Smartphone("iPhone",256)
#Agregando los smartphones a una linked list
miListaDeSmartphones=LinkedList()
miListaDeSmartphones.agregar_al_final(tel1)
miListaDeSmartphones.agregar_al_final(tel2)
miListaDeSmartphones.agregar_al_final(tel3)
miListaDeSmartphones.agregar_al_final(tel4)
miListaDeSmartphones.agregar_al_final(tel5)
#Imprimiendo el contenido
miListaDeSmartphones.transversal()
#Eliminando el de la posicion 2
miListaDeSmartphones.eliminar(2)
#Imprimiendo el contenido
print()
miListaDeSmartphones.transversal()
```

```

#Actualizando el segundo elemento
tel6=Smartphone("Oppo",120)
miListaDeSmartphones.actualizar(tel2,tel6)
#Agregando un elemento al inicio
tel7=Smartphone("LG",80)
miListaDeSmartphones.agregar_al_inicio(tel7)
#Agregando un elemento al final
tel8=Smartphone("Poco",256)
miListaDeSmartphones.agregar_al_final(tel8)
#Imprimiendo el contenido
print()
miListaDeSmartphones.transversal()
#Eliminando el primero
miListaDeSmartphones.eliminar_el_primer()
#Imprimiendo el contenido
print()
miListaDeSmartphones.transversal()

```

---

Ejecuciones del *main*:

```

|Marca: Nokia, Capacidad: 120GB| --> |Marca: Xiamoi, Capacidad: 256GB| --> |Marca: Samsung, Capacidad: 500GB| --> |Marca: Motorola, Capacidad: 80GB| --> |Marca: iPhone, Capacidad: 256GB| --> None

```

```

|Marca: Nokia, Capacidad: 120GB| --> |Marca: Xiamoi, Capacidad: 256GB| --> |Marca: Motorola, Capacidad: 80GB| --> |Marca: iPhone, Capacidad: 256GB| --> None

```

```

|Marca: LG, Capacidad: 80GB| --> |Marca: Nokia, Capacidad: 120GB| --> |Marca: Oppo, Capacidad: 120GB| --> |Marca: Motorola, Capacidad: 80GB| --> |Marca: iPhone, Capacidad: 256GB| --> |Marca: Poco, Capacidad: 256GB| --> None

```

```

|Marca: Nokia, Capacidad: 120GB| --> |Marca: Oppo, Capacidad: 120GB| --> |Marca: Motorola, Capacidad: 80GB| --> |Marca: iPhone, Capacidad: 256GB| --> |Marca: Poco, Capacidad: 256GB| --> None

```