

PRÁCTICO LISTAS

```
from linkedlist import *
from algo1 import *

"""
1- Dadas las siguientes listas de elementos:
A) 24, 45, 3, 67, 89, 345, 54, 22, 3, 678
B) 46, 34, 64, 59, 12, 15, 234, 567, 12, 33

Llenar dos listas A y B con los elementos dados, e imprimir su
contenido.
"""

def mostrarLinkedListConTexto(L, t):
    current = L.head
    print(t + ": [", end="")
    while current != None:
        if current.nextNode != None:
            print(str(current.value) + ", ", end="")
        else:
            print(str(current.value), end="")
        current = current.nextNode
    print("]")

A = LinkedList()
B = LinkedList()
#A.head = None
#B.head = None
add(A, 678)
add(A, 3)
add(A, 22)
add(A, 54)
add(A, 345)
add(A, 89)
add(A, 67)
add(A, 3)
add(A, 45)
add(A, 24)
mostrarLinkedListConTexto(A, "A")
```

```
add(B, 33)
add(B, 12)
add(B, 567)
add(B, 234)
add(B, 15)
add(B, 12)
add(B, 59)
add(B, 64)
add(B, 34)
add(B, 46)
mostrarLinkedListConTexto(B, "B")

"""
Crear una lista C e intercalar los elementos de las dos listas
anteriores, imprimir su contenido.
"""
C = LinkedList()

def intercalarListas(A, B):
    currentNodeA = A.head
    currentNodeB = B.head

    c = 0

    while currentNodeA != None and currentNodeB != None:
        insert(C, currentNodeA.value, c)
        currentNodeA = currentNodeA.nextNode
        c += 1
        insert(C, currentNodeB.value, c)
        currentNodeB = currentNodeB.nextNode
        c += 1

intercalarListas(A,B)
mostrarLinkedListConTexto(C, "C LISTA INTERCALADA")

"""
Buscar en la lista C todos los elementos pares de la lista A y
eliminarlos, imprimir la lista C final.
"""
def eliminaPares(L):
    currentNode = L.head
```

```
while currentNode != None:
    if (currentNode.value % 2 == 0):
        delete(L, currentNode.value)
    currentNode = currentNode.nextNode

eliminaPares(C)
mostrarLinkedListConTexto(C, "C LISTA DE PARES")

"""
Generar una lista D con todos los elementos impares de C, finalmente
imprimir el contenido de D.
"""

def generaListaDeImpares(L, D):
    currentNode = L.head

    while currentNode != None:
        if (currentNode.value % 2 != 0):
            add(D, currentNode.value)
        currentNode = currentNode.nextNode

D = LinkedList()

generaListaDeImpares(C, D)
mostrarLinkedListConTexto(D, "D LISTA DE IMPARES")

"""
Quitarle a la lista A los elementos repetidos y añadirle al final todos
los elementos de B que se encuentren entre 50 y 100, imprimir la lista
A resultante.
"""

def quitarRepetidos(L):
    currentNode = L.head

    while currentNode != None:
        #print("currentNode: " + str(currentNode.value))
        postNode = currentNode.nextNode
        while postNode != None:
            #print("postNode: " + str(postNode.value))
            if (currentNode.value == postNode.value):
                delete(L, postNode.value)
```

```
        postNode = postNode.nextNode

        currentNode = currentNode.nextNode

def aniadeElementosDeBAFinal(L):
    c = length(L)
    currentNodeB = B.head

    while currentNodeB != None:

        if currentNodeB.value >= 50 and currentNodeB.value <= 100:
            insert(L,currentNodeB.value,c)
            c += 1

        currentNodeB = currentNodeB.nextNode

aniadeElementosDeBAFinal(A)
mostrarLinkedListConTexto(A, "A CON ELEMENTOS ENTRE 50 Y 100 DE B")

"""
2- Desarrolle un algoritmo en pseudocódigo que permita cargar una lista
en donde su campo "value" sea igual a una estructura "Empleado" que
tenga tres campos: "nombre", "edad" y "nroLegajo".

Cargar la lista de empleados.
Imprimir la lista cargada.

Empleados:
    Eduardo Ángel, 34, 2
    Juan Carlos, 23, 5
    Luis Esteban, 32, 7
    Juan Carlos, 23, 5
    Pedro Augusto, 40, 9
    Luis Esteban, 32, 7
    Pedro César, 45, 8
    Eduardo Ángel, 34, 2
    Luis Esteban, 32, 7

"""
class Empleado:
```

```
nombre = None
edad = None
nroLegajo = None

def mostrarEmpleados(L):
    current = L.head
    print("---- EMPLEADOS: ")
    while current != None:
        if current.nextNode != None:
            print("----- " + current.value.nombre + "-" +
str(current.value.edad) + "-" + str(current.value.nroLegajo) + ", ")
        else:
            print("----- " + current.value.nombre + "-" +
str(current.value.edad) + "-" + str(current.value.nroLegajo))
        current = current.nextNode

def addEmpleado(L, nombreIn, edadIn, nroLegajoIn):
    newNode = Node()
    newNode.value = Empleado()

    newNode.value.nombre = nombreIn
    newNode.value.edad = edadIn
    newNode.value.nroLegajo = nroLegajoIn

    newNode.nextNode = L.head
    L.head = newNode

Empleados = LinkedList()
addEmpleado(Empleados, "Luis Esteban", 32, 7)
addEmpleado(Empleados, "Eduardo Ángel", 34, 2)
addEmpleado(Empleados, "Pedro César", 45, 8)
addEmpleado(Empleados, "Luis Esteban", 32, 7)
addEmpleado(Empleados, "Pedro Augusto", 40, 9)
addEmpleado(Empleados, "Juan Carlos", 23, 5)
addEmpleado(Empleados, "Luis Esteban", 32, 7)
addEmpleado(Empleados, "Juan Carlos", 23, 5)
addEmpleado(Empleados, "Eduardo Ángel", 34, 2)

mostrarEmpleados(Empleados)
```

```
"""
3- Para el ejercicio anterior:

Eliminar los elementos donde "nroLegajo" se encuentren duplicados
Agregar antes del legajo número 7 el siguiente: Ernesto Andrés, 55, 6
Mover el legajo 9 luego del legajo 8
Imprimir la lista resultante

"""
def eliminarLegajoDuplicado(L):
    currentEmpleado = L.head

    while currentEmpleado != None:
        #print("currentEmpleado: " + str(currentEmpleado.value.nombre))
        postEmpleado = currentEmpleado.nextNode
        while postEmpleado != None:
            #print("postEmpleado: " + str(postEmpleado.value.nombre))
            if (currentEmpleado.value.nroLegajo ==
postEmpleado.value.nroLegajo):
                delete(L, postEmpleado.value)
                postEmpleado = postEmpleado.nextNode

            currentEmpleado = currentEmpleado.nextNode

eliminarLegajoDuplicado(Empleados)
mostrarEmpleados(Empleados)

def insertEmpleado(L, newEmpleadoNode, position):

    if position <= length(L) and position >= 0:
        if position == 0:
            add(L, newEmpleadoNode)
        else:
            currentNode = L.head
            for i in range(0, position - 1):
                currentNode = currentNode.nextNode

            newEmpleadoNode.nextNode = currentNode.nextNode
            currentNode.nextNode = newEmpleadoNode
            return position

    return None
```

```
def mueveYAgregaEmpleados(L, newEmpleadoNode) :
    insertEmpleado(L,newEmpleadoNode, 2)

    nodeToMove = access(L,4)
    delete(L, nodeToMove)
    empleadoAMover = Node()
    empleadoAMover.value = nodeToMove

    insertEmpleado(L,empleadoAMover, 5)
```

```
newEmpleado = Node()
newEmpleado.value = Empleado()
newEmpleado.value.nombre = "Ernesto Andrés"
newEmpleado.value.edad = 55
newEmpleado.value.nroLegajo = 6
```

```
mueveYAgregaEmpleados(Empleados, newEmpleado)
mostrarEmpleados(Empleados)
```

```
"""
```

4- Desarrolle un algoritmo que ordene de manera inversa los elementos (nodos) de una lista.

```
"""
```

```
L = LinkedList()
add(L, 3)
add(L, 2)
add(L, 1)
add(L, 4)
mostrarLinkedList(L)
```

```
def invierteLista(L):
    previousNode = None
    currentNode = L.head
    temporalNode = None

    while currentNode != None:
        temporalNode = currentNode.nextNode
```

```
currentNode.nextNode = previousNode
previousNode = currentNode
currentNode = temporalNode
L.head = previousNode
```

```
invierteLista(L)
```

```
mostrarLinkedList(L)
```

```
A: [24, 45, 3, 67, 89, 345, 54, 22, 3, 678]
B: [46, 34, 64, 59, 12, 15, 234, 567, 12, 33]
C LISTA INTERCALADA: [24, 46, 45, 34, 3, 64, 67, 59, 89, 12, 345, 15, 54, 234, 22, 567, 3, 12, 678, 33]
C LISTA DE PARES: [45, 3, 67, 59, 89, 345, 15, 567, 3, 33]
D LISTA DE IMPARES: [33, 3, 567, 15, 345, 89, 59, 67, 3, 45]
A CON ELEMENTOS ENTRE 50 Y 100 DE B: [24, 45, 3, 67, 89, 345, 54, 22, 3, 678, 64, 59]
---- EMPLEADOS:
----- Eduardo Ángel-34-2,
----- Juan Carlos-23-5,
----- Luis Esteban-32-7,
----- Juan Carlos-23-5,
----- Pedro Augusto-40-9,
----- Luis Esteban-32-7,
----- Pedro César-45-8,
----- Eduardo Ángel-34-2,
----- Luis Esteban-32-7
---- EMPLEADOS:
----- Eduardo Ángel-34-2,
----- Juan Carlos-23-5,
----- Luis Esteban-32-7,
----- Pedro Augusto-40-9,
----- Pedro César-45-8
---- EMPLEADOS:
----- Eduardo Ángel-34-2,
----- Juan Carlos-23-5,
----- Ernesto Andrés-55-6,
----- Luis Esteban-32-7,
----- Pedro César-45-8,
----- Pedro Augusto-40-9
LINKED LIST: [4, 1, 2, 3]
LINKED LIST: [3, 2, 1, 4]
```