De Faveri Tomas

```
from linkedlist import *
from algo1 import *
from myarray import *
import random

"""
Desarrolle un algoritmo en pseudopython que permita generar de manera
totalmente aleatoria una lista de 15 nodos en donde su campo “value”
sea igual a una estructura “Empleado” que tenga los siguientes campos:
“nombre”, “edad”, “nroLegajo” y “fechaIngreso” donde este último esté
compuesto de: “dia”, “mes” y “anio”.

Premisas:
-el campo Nombre será elegido aleatoriamente tomado de un array de 10
nombres precargados:
nombres: “Juan”, “Pedro”, “Laura”, “Ana”, …
-edad se llenará aleatoriamente entre 20 y 70 años.
-nroLegajo será un número aleatorio entre 1 y 1000 pero asegurándose de
que sea sin generar duplicados. (ya sea recorriendo los nodos ya
generados o con ayuda de un arreglo o lista extra “nrosGenerados”)
-fechaIngreso serán dia (aleat. 1 al 31), mes (aleat. 1 al 12) y anio
será un aleatorio tal que dependiendo de la edad del empleado no
hubiera ingresado antes de sus 20 años…
    por ejemplo: si el empleado tiene 22 años su fecha de ingreso
estará entre 2021 y 2023… si tiene 20 años sólo podrá haber ingresado
en el 2023…

"""


class Fecha:
    dia = None
    mes = None
    anio = None

class Empleado:
    nombre = None
    edad = None
    nroLegajo = None
    fechaIngreso = Fecha()


def muestraEmpleados(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) + "-(" +
str(current.value.fechaIngreso.dia) + "-"+
str(current.value.fechaIngreso.mes) + "-" +
str(current.value.fechaIngreso.anio) + "), ")
        else:
            print("------ " + current.value.nombre + "-" +
str(current.value.edad) + "-" + str(current.value.nroLegajo) + "-(" +
str(current.value.fechaIngreso.dia) + "-"+
str(current.value.fechaIngreso.mes) + "-" +
str(current.value.fechaIngreso.anio) + ")")
        current = current.nextNode


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

    newNode.value.nombre = nombreIn
    newNode.value.edad = edadIn
    newNode.value.nroLegajo = nroLegajoIn
    newNode.value.fechaIngreso = Fecha()
    newNode.value.fechaIngreso.dia = diaIn
    newNode.value.fechaIngreso.mes = mesIn
    newNode.value.fechaIngreso.anio = anioIn

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


def generaLista(E, arrNombres):

    arrLegajos = Array(14, 0)

    for n in range(0,15):
        bool = True

        nombreEmpleado = arrNombres[random.randint(0,9)]
```

```
        edadEmpleado = random.randint(20,70)

        while bool:
            nroLegajoEmpleado = random.randint(1,1000)
            if search(arrLegajos, nroLegajoEmpleado) == None:
                insert(arrLegajos, nroLegajoEmpleado, n)
                bool = False

        diaIngresoEmpleado = random.randint(1,31)
        mesIngresoEmpleado = random.randint(1,12)

        """
        20 = 2025
        21 = 2025-2024
        22 = 2025-2024-2023
        23 = 2025-2024-2023-2022
        """
        limiteBajo = 2025 - (edadEmpleado - 20)
        anioIngresoEmpleado = random.randint(limiteBajo,2025)

        addEmpleado(E, nombreEmpleado, edadEmpleado, nroLegajoEmpleado,
diaIngresoEmpleado, mesIngresoEmpleado, anioIngresoEmpleado)


def buscaLegajoMenor(E):
    currentNode = E.head
    legajoMenor = E.head

    while currentNode != None:
        if currentNode.value.nroLegajo < legajoMenor.value.nroLegajo:
            legajoMenor = currentNode

        currentNode = currentNode.nextNode

    #print("MENOR " + legajoMenor.value.nombre + "-" +
str(legajoMenor.value.edad) + "-" + str(legajoMenor.value.nroLegajo) +
"-(" + str(legajoMenor.value.fechaIngreso.dia) + "-"+
str(legajoMenor.value.fechaIngreso.mes) + "-" +
str(legajoMenor.value.fechaIngreso.anio) + ")")

    return legajoMenor

def searchLegajo(L, legajo):
```

```
    current = L.head
    position = 0
    while current != None:
        if current.value.nroLegajo == legajo:
            return position
        current = current.nextNode
        position += 1

    return None


def eliminaEmpleadoXLegajo(E, nodo):

    position = searchLegajo(E, nodo.value.nroLegajo)

    if position != None:

        if position == 0:
            E.head = E.head.nextNode
        else:
            currentNode = E.head
            for i in range(0, position - 1):
                currentNode = currentNode.nextNode
            currentNode.nextNode = currentNode.nextNode.nextNode

        return position
    else:
        return None


def eliminarLegajosMenores(E):

    for n in range(0,5):
        nodoAEliminar = buscaLegajoMenor(E)
        eliminaEmpleadoXLegajo(E, nodoAEliminar)


"""
Generar la lista de 15 empleados aleatoriamente.
Imprimir en pantalla la lista generada mediante el uso de una función
muestraEmpleados() que muestre un registro de cada empleado con todos
sus datos.
eliminar los 5 empleados de menor número de legajo.
```

De Faveri Tomas

```
volver a imprimir en pantalla la lista de Empleados final.
"""

Empleados = LinkedList()
arrNombres =
["Juan","Pedro","Laura","Ana","Pepe","Renzo","Tomas","Julio","Pablo","M
artin"]

generaLista(Empleados, arrNombres)
muestraEmpleados(Empleados)
eliminarLegajosMenores(Empleados)
muestraEmpleados(Empleados)
```

```
---- EMPLEADOS:
------ Pedro-57-915-(8-7-2005),
------ Ana-50-466-(5-10-2022),
------ Renzo-36-69-(13-2-2014),
------ Renzo-60-643-(31-11-2013),
------ Pedro-66-556-(11-10-1995),
------ Ana-23-464-(26-12-2023),
------ Laura-37-841-(12-6-2025),
------ Pedro-70-678-(25-9-1985),
------ Laura-26-394-(21-8-2022),
------ Renzo-60-516-(15-2-2006),
------ Renzo-38-791-(29-5-2008),
------ Pedro-23-414-(3-4-2025),
------ Renzo-23-222-(16-9-2025),
------ Pablo-40-647-(6-11-2024),
------ Pablo-69-338-(19-6-2012)
---- EMPLEADOS:
------ Pedro-57-915-(8-7-2005),
------ Ana-50-466-(5-10-2022),
------ Renzo-60-643-(31-11-2013),
------ Pedro-66-556-(11-10-1995),
------ Ana-23-464-(26-12-2023),
------ Laura-37-841-(12-6-2025),
------ Pedro-70-678-(25-9-1985),
------ Renzo-60-516-(15-2-2006),
------ Renzo-38-791-(29-5-2008),
------ Pablo-40-647-(6-11-2024)
```