

Auxiliatura INF-131 “C”

Estructura de Datos y Algoritmos

Univ. Miguel Angel Quispe Mamani

Universidad Mayor de San Andrés

Carrera de Informática

24/04/2023

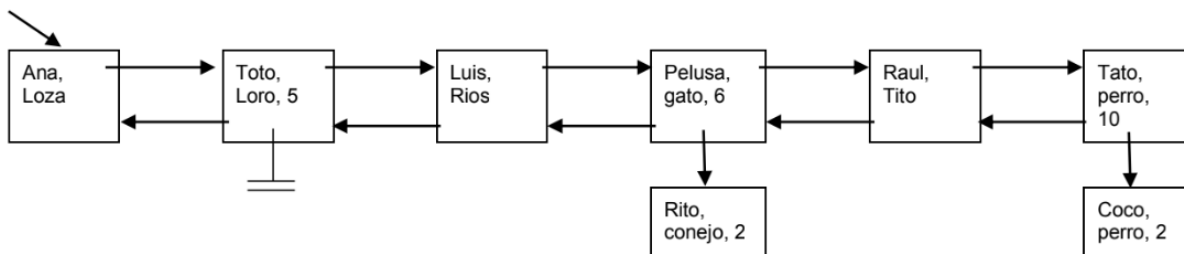
Nota: Las soluciones de este examen pasado estan en el lenguaje [SeudoCodigo](#)

Enunciado

Puede verlo en el siguiente enlace: [Examen 2do. Parcial II/2021](#)

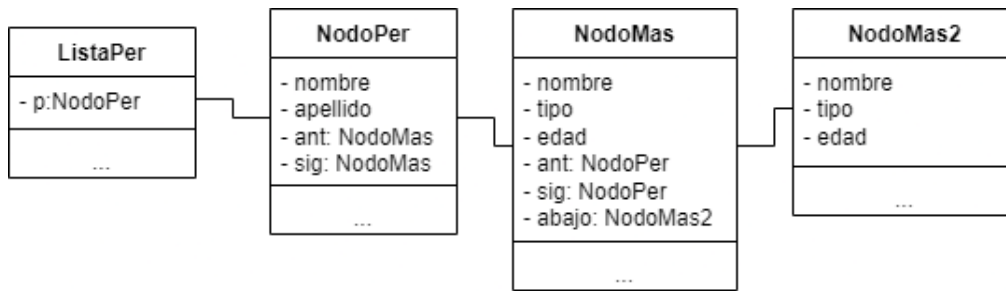
Ejercicio 2

Sea una lista que controla datos de dueños(nombre,apellido) que pueden tener 1 o 2 mascotas(nombre,tipo,edad) de la siguiente manera:



- (5pts) Elaborar el diagrama de clases
- (5pts) Mostrar a las mascotas con la mayor edad y a su dueño

Inciso a



Inciso b

```

1 mayorEdad(ListaPer a){//devuelve la edad maxima de una mascota
2   w = a.getP() //NodoPer
3   mayor = 0
4   while w <> null{
5     v = w.getSig() //NodoMas, primera mascota
6     if v.getEdad() > mayor
7       mayor = v.getEdad()
8     //obtener la mascota de abajo
9     x = v.getAbajo() // NodoMas2
10    if x <> null{
11      if x.getEdad() > mayor
12        mayor = x.getEdad()
13    }
14    w = w.getSig().getSig()
15  }
16  return mayor
17 }

```

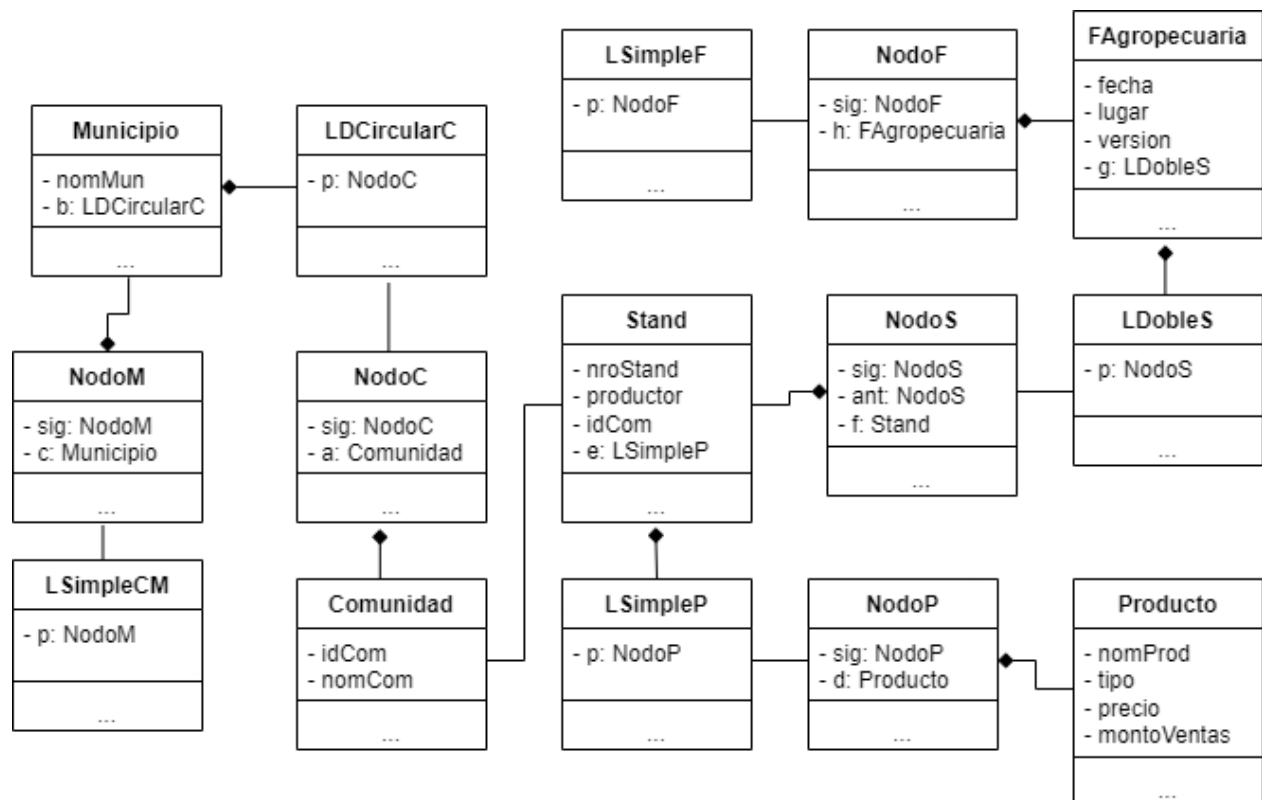
```

1 mostrarMayorEdad(ListaPer a){
2   maximo = mayorEdad(a)
3   w = a.getP() //NodoPer
4   while w <> null{
5     v = w.getSig() //NodoMas, primera mascota
6     if v.getEdad() = maximo{
7       print("mascota:", v.getNombre(), v.getTipo(), v.getEdad())
8       print("duenio: ", w.getNombre(), w.getApellido())
9     }
10    //obtener la mascota de abajo
11    x = v.getAbajo() // NodoMas2
12    if x <> null{
13      if x.getEdad() = maximo{
14        print("mascota:", x.getNombre(), x.getTipo(), x.getEdad())
15        print("duenio: ", w.getNombre(), w.getApellido())
16      }
17    }
18    w = w.getSig().getSig()
19  }
20  return mayor
21 }

```

Considerando que la información de las ferias agropecuarias de La Paz se almacena en las siguientes estructuras:

- Inciso a



Inciso b

```

1 devolver(LSimpleCM a, String idCom){//Devuelve el nomMuni, dado un IdCom
2     w = a.getP() //NodoM
3     while w.getSig() <> a.getP(){
4         m = w.getC() //Municipio
5         b = m.getB() //LDCircularC
6         w2 = b.getP() // NodoC
7         while w2.sig() <> b.getP(){
8             c = w2.getA() //comunidad
9             if c.getIdCom() = idCom
10                return m.getNomMun()
11            w2 = w2.getSig()
12        }
13        c = w2.getA() //comunidad
14        if c.getIdCom() = idCom
15            return m.getNomMun()
16        w = w.getSig()
17    }
18    m = w.getC() //Municipio
19    b = m.getB() //LDCircularC
20    w2 = b.getP() // NodoC
21    while w2.sig() <> b.getP(){
22        c = w2.getA() //comunidad
23        if c.getIdCom() = idCom
24            return m.getNomMun()
25        w2 = w2.getSig()
26    }
27    c = w2.getA() //comunidad
28    if c.getIdCom() = idCom
29        return m.getNomMun()
30    return "no existe tal comunidad"
31 }

```

```

1 IncisoB(LSimpleCM a, LSimpleF b, int i, String municipioX){
2     w = b.getP() //NodoF
3     cont = j = 0
4     while w <> null{
5         j = j + 1
6         if j = i{
7             fa = w.getH() // feriaAgrop
8             c = fa.getG()//LDobleS
9             w2 = c.getP() //NodoS
10            while w2 <> null{
11                st = w2.getF() //Stand
12                esteMunicipio = devolver(a, st.getIdCom())
13                if esteMunicipio = municipioX
14                    cont = cont + 1
15                w2 = w2.getSig()
16            }
17        }
18        w = w.getSig()
19    }
20    print("Hay ", cont, " comunidades de dicho municipio")
21 }

```

Inciso c

```

1 IncisoC(LSimpleF a, int i, int x){
2     w = a.getP() //NodoF
3     j = 0
4     while w <> null{
5         j = j + 1
6         if i = j{
7             fa = w.getH() //feria agro
8             b = fa.getG() //LDobleS
9             w2 = b.getP()//NodoS
10            while w2 <> null{
11                st = w2.getF() //stand
12                c = st.getE() // LSimpleP
13                w3 = c.getP() //NodoP
14                n = c.nroNodos()
15                for ii = 1 to n{
16                    w3 = c.getP()
17                    //si el primer nodo tiene precio > x
18                    if w3.getD().getPrecio() > x{
19                        c.eliPrincipio() //a implementar
20                    }else{
21                        sw = false //no hay precio > x
22                        while w3 <> null{
23                            pr = w3.getD()//Producto
24                            if pr.getPrecio() > x{
25                                sw = true
26                                break
27                            }
28                            w3 = w3.getSig()
29                        }
30                        if sw == true{
31                            w4 = c.getP()//NodoP
32                            while(w4.getSig() != w3)
33                                w4 = w4.getSig()
34                            w4.setSig(w3.getSig())
35                        }
36                    }
37                }
38                w2 = w2.getSig()
39            }
40        }
41        w = w.getSig()
42    }
43 }

```

Nota: A continuación se presenta la implementación de los metodos faltantes para el inciso c: [nroNodos](#) y [eliPrincipio](#)

```
1 class LSimpleP{
2     ...
3     nroNodos() {
4         w = getP() //NodoP
5         c = 0
6         while w <> null {
7             c = c + 1
8             w = w.getSig()
9         }
10        return c
11    }
12    ...
13    eliPrincipio() {
14        r = getP() //NodoP
15        setP(r.getSig())
16        r.setSig(null)
17        return r
18    }
19    ...
20 }
```