Tarea inf-121

- Nombre: Freddy Erick Velarde Silva

`Principal` .java:

```
public class Principal {
  public static void main(String[] args) {
    LS_CircularP a = new LS_CircularP();
    // a.llenar(5);
    a.adiFinal("erick", 21);
    a.adiFinal("freddy", 22);
    a.adiFinal("alejandro", 20);
    a.adiFinal("pedro", 20);
    a.adiFinal("Camila", 20);
        a.mostrar();

        NodoP z = a.eliPrincipio();
    System.out.println("first elemente: " + z.getNom() + " " + z.getEdad());

    a.adiFinal("Marcelo", 25);
    a.mostrar();
  }
}
```

output:

> Agregar 5 personas:

```
javac -d bin -sourcepath src sr
Nombre: erick edad: 21
Nombre: freddy edad: 22
Nombre: alejandro edad: 20
Nombre: pedro edad: 20
Nombre: Camila edad: 20
```

> Eliminar el primer elemento:

```
javac -d bin -sourcepath
first element: erick 21
```

> Agregar a Marcelo:

```
javac -d bin -sourcepath src
Nombre: freddy edad: 22
Nombre: alejandro edad: 20
Nombre: pedro edad: 20
Nombre: Camila edad: 20
Nombre: Marcelo edad: 25
```

source code: `NodoP` .java:

```java
public class NodoP {
  private String nom;
  private int edad;
  private NodoP sig;
```

```
    public NodoP() { this.sig = null; }

    public String getNom() { return nom; }
    public void setNom(String nom) { this.nom = nom; }
    public int getEdad() { return edad; }
    public void setEdad(int edad) { this.edad = edad; }
    public void setSig(NodoP sig) { this.sig = sig; }
    public NodoP getSig() { return sig; }
  }
```

ListaSimpleP .java

```
  public class ListaSimpleP {
    protected NodoP P;

    public ListaSimpleP() { P = null; }

    public NodoP getP() { return P; }

    public void setP(NodoP p) { P = p; }
  }
```

LS_CircularP .java

```
  import java.util.Scanner;

  public class LS_CircularP extends ListaSimpleP {
    private int size = 0;
    public LS_CircularP() { super(); }

    public boolean esVacia() { return this.P == null; }

    public int nroNodos() { return size; }
```

```
public void adiFinal(String nom, int edad) {
  NodoP newNode = new NodoP();
  newNode.setNom(nom);
  newNode.setEdad(edad);

  if (this.P == null) {
    P = newNode;
    P.setSig(P);
  } else {
    NodoP currentNode = P;
    while (currentNode.getSig() != P) {
      currentNode = currentNode.getSig();
    }
    currentNode.setSig(newNode);
    newNode.setSig(P);
  }
  size++;
}

public void mostrar() {
  NodoP currentNode = P;
  if (P == null) {
    System.out.println("lista vacia");
    return;
  }

  while (currentNode.getSig() != P) {
    System.out.println("Nombre: " + currentNode.getNom() +
                       " edad: " + currentNode.getEdad());
    currentNode = currentNode.getSig();
  }
  System.out.println("Nombre: " + currentNode.getNom() +
                     " edad: " + currentNode.getEdad());
}
```

```
public void adiPrincipio(String nom, int edad) {
  NodoP newNode = new NodoP();
  newNode.setNom(nom);
  newNode.setEdad(edad);

  if (P == null) {
    P = newNode;
    P.setSig(P);
  } else {
    NodoP currentNode = P;
    while (currentNode.getSig() != P) {
      currentNode = currentNode.getSig();
    }
    currentNode.setSig(newNode);
    newNode.setSig(P);
    P = newNode;
    size++;
  }
}

public NodoP eliFinal() {
  NodoP node_a = new NodoP();
  if (P != null) {
    if (P.getSig() == P) {
      node_a = P;
      node_a.setSig(P);
      P = null;
    } else {
      NodoP node_b = new NodoP();
      NodoP currentNode = P;
      while (currentNode.getSig() != P) {
        node_b = currentNode;
        currentNode = currentNode.getSig();
      }

      node_a = currentNode;
```

```
        node_a.setSig(null);
        node_b.setSig(P);
      }
    }
    size--;
    return node_a;
  }

  public NodoP eliPrincipio() {
    NodoP node = new NodoP();
    if (esVacia()) {
      System.out.println("lista vacia");
      return node;
    }

    if (size == 1) {
      P = null;
      size--;
    } else {
      NodoP temp = P;
      node = P;
      while (temp.getSig() != P) {
        temp = temp.getSig();
      }
      temp.setSig(P.getSig());
      P = P.getSig();
      size--;
    }
    return node;
  }

  public void llenar1(int n) {
    Scanner scanner = new Scanner(System.in);

    for (int i = 0; i < n; i++) {
      System.out.println("Insert: nom, edad: ");
```

```
        String nom = scanner.next();
        int edad = scanner.nextInt();

        adiFinal(nom, edad);
      }
    }
    public void llenar2(int n) {
      Scanner reader = new Scanner(System.in);
      String nom;
      int edad;

      for (int i = 1; i <= n; i++) {
        System.out.println("Insert: nom, edad:");
        nom = reader.next();
        edad = reader.nextInt();

        adiPrincipio(nom, edad);
      }
    }
  }
```