```
1
2   public class incremento {
3
4       public incremento();
5
6       public void pseudocodigo(Vecino jose) {
7           SistemaDeRegistro sr;
8           SistemaDeTelefono t;
9           LectorDeCodigo l;
10          AdminPuntoLimpio p;
11
12
13          if (sr.quiereRegistrar(jose)) {
14              List<PuntoLimpio> lp = damePL(codigoDeBarras);
15              // Suposición: El sistema ya almacenó el codigo de barras.
16              // Aquí elige un Punto Limpio.
17              PuntoLimpio pl;
18              // Aquí ingresa la cantidad que desea reciclar.
19              pl.agregarReciclable(codigoDeBarras, cant);
20              jose.registrarResiduo(codigoDeBarras, cant);
21          }
22      }
23
24
25  /* Dentro de la clase Sistema de Registro */
26
27  public bool quiereRegistrar(Vecino v){
28      Image i = t.solicitarImagen();  }
29      int codigoDeBarras = l.transformar(i);
30      boolean b = p.verificarCodigoDeBarras(codigoDeBarras);
31      return b;
32  }
33
34  public List<PuntoLimpio> damePL(int codigoDeBarras){
35      return p.dondePuedoReciclar(codigoDeBarras);
36  }
37
38  /* Dentro de la clase AdminPuntoLimpio */
39
40  HashMap<int,List<PuntoLimpio>> dondePuedoReciclar;
41
42  public List<PuntoLimpio> dondePuedoReciclar(int codigoDeBarras){
43      List<PuntoLimpio> lp = new ArrayList<PuntoLimpio>();
44      lp.addAll(dondePuedoReciclar.get(codigoDeBarras));
45      return lp;
46  }
47
48  /* Dentro de la clase Punto Limpio */
49
50  HashMap<int,int> almacenado;
51
52  public void agregarReciclable(int codigoDeBarras, int cant) {
53      int alm = almacenado.get(codigoDeBarras);
54      cant += alm;
55      almacenado.put(codigoDeBarras,cant);
56  }
57
58  /* Dentro de la clase Vecino */
59
60  HashMap<int,int> reciclados;
61
62  public void registrarResiduo(int codigoDeBarras, int cant) {
63      int alm = reciclados.get(codigoDeBarras);
64      cant += alm;
65      reciclados.put(codigoDeBarras,cant);
66  }
```