



13:50

MonitorCPU.java

```
27     }
28
29     double consumo;
30
31     try {
32         consumo = Double.parseDouble(entrada);
33     } catch (NumberFormatException e) {
34         System.out.println("[Error] Entrada no válida, debe ser un número.");
35         continue;
36     }
37
38     if (consumo < 0 || consumo > 100) {
39         System.out.println("[Error] El consumo debe estar entre 0 y 100.");
40         continue;
41     }
42
43     if (consumosRegistrados.contains(consumo)) {
44         System.out.println("[Aviso] Consumo duplicado, no será registrado.");
45         continue;
46     }
47
48     if (consumo > 95) {
49         throw new ConsumoCriticoException("Consumo crítico detectado: " + consumo + "%");
50     }
51
52     consumosRegistrados.add(consumo);
53     System.out.println("[Consumo] " + consumo + "% registrado correctamente.");
```

Home End → ( ) { } < > ' " ;

✎ 📁 ▶ 📄

13:50

MonitorCPU.java

```
48         if (consumo > 95) {
49             throw new ConsumoCriticoException("Consumo crítico detectado: " + consumo + "%");
50         }
51
52         consumosRegistrados.add(consumo);
53         System.out.println("Consumo " + consumo + "% registrado correctamente.");
54     }
55
56     System.out.println("\nRegistro finalizado. Consumos registrados:");
57     for (Double c : consumosRegistrados) {
58         System.out.println("- " + c + "%");
59     }
60
61 } catch (ConsumoCriticoException e) {
62     System.out.println("[Alerta crítica] " + e.getMessage());
63 } catch (Exception e) {
64     System.out.println("[Error inesperado] " + e.getMessage());
65 } finally {
66     if (scanner != null) {
67         scanner.close();
68         System.out.println("Recurso Scanner cerrado.");
69     }
70 }
71 }
72 }
73 }
```

Home End → ( ) { } < > ' " ;

🔍 📁 ▶ 🖨

Registro finalizado. Consumos registrados:  
Recurso Scanner cerrado.