

PuestoPerritos.java

```
1 import java.util.Random;
2
3 /**
4  * Clase que modela un puesto que vende perritos calientes.
5  *
6  *
7  */
8 public class PuestoPerritos {
9
10     private int numPerritos;
11     private int numPanes;
12     private int numSalchichas;
13     private int numPerritosVendidos;
14     private int reponerPanes;
15     private int reponerSalchichas;
16
17     //constructor
18     /**
19      * Constructor de la clase.
20      *
21      * Precondición: El número de panes y de salchichas debe ser mayor que 0.
22      *
23      * @param numPanesInicial  número de panes.
24      * @param numSalchichasInicial número de salchichas
25      */
26     public PuestoPerritos (int numPanesInicial, int numSalchichasInicial) {
27
28         this.numPanes = numPanesInicial;
29         this.numSalchichas = numSalchichasInicial;
30     }
31
32     /**
33      * Obtiene el número de panes que quedan en el puesto.
34      *
35      * @return número de panes que hay en el puesto.
36      */
37
38     public int getNumPanes() {
39         return numPanes;
40     }
41
42     /**
43      * Obtiene el número de salchichas que quedan en el puesto.
44      *
45      * @return número de salchichas que hay en el puesto.
46      */
47
48     public int getNumSalchichas() {
49         return numSalchichas;
50     }
51
52     /**
53      * Obtiene el número de perritos que se pueden hacer con lo que hay en el puesto.
54      *
55      * @return número de perritos que se pueden hacer con lo que hay en el puesto.
56      */
57
58     public int getNumeroPerritos(){
59
60         if (numPanes >= numSalchichas) {
61             numPerritos = numSalchichas;
62         }else {
```

# PuestoPerritos.java

```

63         numPerritos = numPanes;
64     }
65     return numPerritos;
66 }
67
68 /**
69  * Vende perritos
70  * Si no hay suficientes perritos se venden los que hay.
71  *
72  * Precondici3n: El n3mero de perritos solicitado debe ser mayor que 0.
73  *
74  * @param numPerritosSolicitados N3mero de perritos que piden los clientes.
75  * @return el n3mero de perritos vendidos.
76  */
77
78 public int venderPerritos(int numPerritosSolicitados) {
79
80     Random perritos = new Random ();
81     numPerritosVendidos = perritos.nextInt (numPerritosSolicitados);
82
83     return numPerritosVendidos;
84 }
85
86 /**
87  * Repone panes.
88  * Precondici3n: El n3mero de panes debe ser mayor que 0
89  *
90  * @param numero n3mero de panes que se a3aden al stock
91  */
92
93 public void reponerPanes(int numero) {
94
95     reponerPanes = numPanes - numero;
96
97     if(reponerPanes == 0) {
98         System.out.println("En este puesto de perritos hay que reponer "+numPanes+ "
99         panes de perritos.");
100     }
101
102     if(reponerPanes !=0 && numero!=0) {
103         System.out.println("En este puesto de perritos hay que reponer "+numero+ "
104         panes de perritos.");
105     }
106
107     if(reponerPanes == numPanes) {
108         System.out.println("En este puesto de perritos no hay que reponer ningun pan de
109         perritos. ");
110     }
111 }
112
113 /**
114  * Repone salchichas
115  * Precondici3n: El n3mero de salchichas debe ser mayor que 0
116  *
117  * @param numero -n3mero de salchichas que se a3aden al stock
118  */
119
120 public void reponerSalchichas(int numero) {
121
122     reponerSalchichas = numSalchichas - numero;
123
124     if(reponerSalchichas == 0) {

```

PuestoPerritos.java

```
122         System.out.println("En este puesto de perritos hay que reponer "+numSalchichas+
    " salchichas.");
123     }
124
125     if(reponerSalchichas !=0 && numero!=0){
126         System.out.println("En este puesto de perritos hay que reponer "+numero+ "
    salchichas.");
127     }
128
129     if(reponerSalchichas == numSalchichas) {
130         System.out.println("En este puesto de perritos no hay que reponer ninguna
    salchicha.");
131     }
132 }
133 }
134
```