

```
1 package Ejercicio04;
2
3 import java.util.Comparator;
4 import java.util.Iterator;
5 import java.util.LinkedList;
6 import java.util.List;
7 import java.util.PriorityQueue;
8
9 public class U08E04 {
10
11     public static void main(String[] args) {
12
13         //variables locales
14         int totalClientes = 0; //total clientes que entran en el dia
15         int cuantosEntran = 0; //n° de clientes que entran por minuto
16         double lambda; //media lambda para el calculo de clientes que pueden entrar cada minuto
17         Cliente c; //para crear clientes
18         int tiempoBusqueda; //para calcular el valor que hay que pasar a tBuscar en el cliente
19         int compras; //para calcular el tiempo de cola de cada cliente
20         double tiempoPagar; //para guardar el tiempo de cola para pagar
21         int productosComprados; //para calcular el valor de compra en el cliente
22         double media;
23         double mediaBusqueda;
24         double mediEnCola;
25         int totalProductosComprados = 0;
26         double totalTiempoBusqueda = 0;
27         double totalEnCola=0;
28         double tiempoPagando=0;
29         double mediaPagando=0;
30         double totalTiempo=0;
31         double totalMinutos=0;
32         double totalCajal=0;
33         double totalCaja2=0;
34
35         //creamos el array con la tasa de clientes
36         int[] tasaClientes = new int[11];
37
38         //rellenamos el array con la tasa de clientes esperada
39         tasaClientes[0] = 25;
40         tasaClientes[1] = 40;
41         tasaClientes[2] = 50;
42         tasaClientes[3] = 65;
43         tasaClientes[4] = 80;
44         tasaClientes[5] = 42;
45         tasaClientes[6] = 18;
46         tasaClientes[7] = 21;
47         tasaClientes[8] = 32;
48         tasaClientes[9] = 40;
49         tasaClientes[10] = 60;
50
51         //creamos las colas
```

```

52 LinkedList<Cliente> colaBuscar = new LinkedList<>();
53 LinkedList<Cliente> caja1 = new LinkedList<>();
54 LinkedList<Cliente> caja2 = new LinkedList<>();
55
56 //creamos el reloj y su minuterero
57 double reloj = 9.0;
58 double minuto = 1.0 / 60; //ritmo de avance del reloj de la simulacion
59
60 while ( reloj <= 20 ){ //inicio del while reloj
61
62     //vemos cuantos clientes entran cada minuto
63     lambda = minuto * tasaClientes[(int)reloj - 9];
64     cuantosEntran = getPoisson(lambda);
65
66     //creamos los clientes de ese minuto
67     for(int i = 1; i <= cuantosEntran; i++){ //inicio for
68         tiempoBusqueda = (int)calculoNormal(10,5.8);
69         tiempoBusqueda = Math.abs(tiempoBusqueda); //me salen valores negativos
70         compras = (int)(tiempoBusqueda * (Math.random() * 2));
71         tiempoPagar = Math.round(2 + compras / 4.0);
72         tiempoPagar = cuadraSegundos(tiempoPagar);
73         c = new Cliente(reloj,tiempoBusqueda, tiempoBusqueda, compras, 0, tiempoPagar, 0 );
74         colaBuscar.add(c);
75         totalProductosComprados = totalProductosComprados + compras; //poner despues?
76         totalTiempoBusqueda = totalTiempoBusqueda + tiempoBusqueda; //total minutos buscando
77         tiempoPagando = tiempoPagando + tiempoPagar;
78         totalTiempo = totalTiempoBusqueda + tiempoPagando;
79     }
80     totalClientes += cuantosEntran;
81
82     //restar 1 al campo tQueda de los clientes de la colaBuscar
83     Iterator it = colaBuscar.iterator();
84     while(it.hasNext()){
85         Cliente aux = (Cliente)it.next();
86         aux.setTQueda(aux.getTQueda()-1);
87     }
88
89     //ordenar los clientes por el tiempo de espera
90     colaBuscar.sort(
91         new Comparator<Cliente>(){
92             public int compare(Cliente c1, Cliente c2){
93                 if(c1.getTQueda() < c2.getTQueda()) return 1;
94                 if(c1.getTQueda() < c2.getTQueda()) return -1;
95                 return 0;
96             }
97         }
98     );
99
100
101     //ver los clientes que tiempo queda es menor o igual que 0
102     Iterator ite = colaBuscar.iterator();

```

```

103     while(ite.hasNext()){
104         Cliente aux = (Cliente)ite.next();
105         if(aux.getTQueda() <= 0){
106             aux.setTCola(reloj);
107             aux.setTQueda(aux.getTCobrar());
108             ite.remove();
109             if(caja1.size() <= caja2.size()) caja1.add(aux);
110             else caja2.add(aux);
111         }
112     }
113
114     //contamos cuanta gente hay en caja
115     totalCaja1 = totalCaja1 + caja1.size();
116     totalCaja2 = totalCaja2 + caja2.size();
117
118
119
120
121
122     //vemos cola de caja 1
123     for(int i = 0; i < caja1.size(); i++){
124         Cliente q = caja1.getFirst();
125         q.setTQueda(q.getTQueda()-1);
126         if(q.getTQueda() <= 0){
127             caja1.removeFirst();
128             q.setTSale(reloj);
129             totalEnCola = totalEnCola + (q.getTSale() - q.getTCola());
130         }
131     }
132
133     //vemos cola de caja 2
134     for(int i = 0; i < caja2.size(); i++){
135         Cliente q = caja2.getFirst();
136         q.setTQueda(q.getTQueda()-1);
137         if(q.getTQueda() <= 0){
138             caja2.removeFirst();
139             q.setTSale(reloj);
140             totalEnCola = totalEnCola + (q.getTSale() - q.getTCola());
141         }
142     }
143
144     //control de tiempo
145     reloj = reloj + minuto;
146     reloj = cuadraSegundos(reloj);
147     totalMinutos++;
148     if(reloj > 20){
149         System.out.println("Se cierran las cajas. Pendientes en caja1 " + caja1.size() + " y en caja 2 " + caja2.size() +
150         " en local " + colaBuscar.size());
151         System.out.println();
152         if(colaBuscar.size() > 0) colaBuscar.removeAll(colaBuscar);
153         if(caja1.size() > 0) caja1.removeAll(caja1);

```

```

153         if(caja2.size() > 0) caja2.removeAll(caja2);
154     }
155 }
156
157 //Impresion por pantalla
158 System.out.println("Local abierto de 9:00 a 20:00");
159 System.out.println("Entran " + totalClientes + " clientes");
160 System.out.print("Total productos comprados: ");
161 System.out.println(totalProductosComprados);
162 media = (double)totalProductosComprados/totalClientes;
163 System.out.printf("Compran %.2f productos media.",media );
164 mediaBusqueda = (double)totalTiempoBusqueda/totalClientes;
165 mediaBusqueda = cuadraSegundos(mediaBusqueda);
166 System.out.printf("\nPasa %.2f minutos buscando productos",mediaBusqueda);
167 /*mediEnCola = totalEnCola/totalClientes;
168 mediEnCola = cuadraSegundos(mediEnCola);
169 System.out.printf("\nPasa %.2f minutos en cola", mediEnCola);*/
170 mediaPagando = tiempoPagando/totalClientes;
171 mediaPagando = cuadraSegundos(mediaPagando);
172 System.out.printf("\nPasa %.2f minutos pagando productos",mediaPagando);
173 System.out.printf("\nTiempo por cliente: COMPRANDO: %.2f", (totalTiempoBusqueda*100)/totalTiempo);
174 System.out.print("%");
175 System.out.printf(" PAGANDO: %.2f ", (tiempoPagando*100)/totalTiempo);
176 System.out.println("%");
177 System.out.printf("Tamaño cola caja 1 %.2f clientes/min", totalCaja1/totalMinutos);
178 System.out.printf("\nTamaño cola caja 2 %.2f clientes/min", totalCaja2/totalMinutos);
179 }
180
181
182
183 private static double calculoNormal(double i, double d) {
184     double r1 = Math.random();
185     double r2 = Math.random();
186     double z1 = Math.sqrt((-2)*Math.log(r1))*Math.sin(2*Math.PI*r2);
187     double n1 = z1*i+d;
188     return n1;
189 }
190
191 private static int getPoisson(double lambda) {
192     double L = Math.exp(-lambda);
193     double p = 1.0;
194     int k = 0;
195     do {
196         k++;
197         p = Math.random();
198     } while (p > L);
199     return k-1;
200 }
201
202 //para cuadrar la hora, ya que a veces sales mas de 60 segundos
203 private static double cuadraSegundos(double dato){

```

```

204
205     String tiempo = String.valueOf(dato);
206     int pos = tiempo.indexOf(".");
207     pos++;
208     char caracter = tiempo.charAt(pos);
209     if (caracter == 54 || caracter == 55 || caracter == 56 || caracter == 57){
210         dato = dato + 1;
211         dato = dato -0.6;
212     }
213
214     return dato;
215 }
216
217
218
219 }
220
221
222
223 class Cliente {
224
225     //variables miembro
226     private double tEntra;
227     private double tBuscar;
228     private double tQueda;
229     private int compra;
230     private double tCola;
231     private double tCobrar;
232     private double tSale;
233
234     //constructor
235     public Cliente(double entra, double busca, double queda, int cuantos, double cola, double cobra, double sale){
236         this.tEntra = entra;
237         this.tBuscar = busca;
238         this.tQueda = queda;
239         this.compra = cuantos;
240         this.tCola = cola;
241         this.tCobrar = cobra;
242         this.tSale = sale;
243     }
244
245     //getters
246     public double getTEntra() { return this.tEntra;}
247     public double getTBuscar() { return this.tBuscar;}
248     public double getTQueda() { return this.tQueda;}
249     public int getCompra() { return this.compra;}
250     public double getTCola() { return this.tCola; }
251     public double getTCobrar() { return this.tCobrar;}
252     public double getTSale() { return this.tSale;}
253
254     //setters

```

```
255     public void setTEntra(double entra) { this.tEntra = entra;}
256     public void setTBuscar(double busca) { this.tBuscar = busca;}
257     public void setTQueda(double queda) { this.tQueda = queda;}
258     public void setCompra(int cantidad) { this.compra = cantidad;}
259     public void setTCola(double cola) { this.tCola = cola; }
260     public void setTCobrar(double cobra) { this.tCobrar = cobra;}
261     public void setTSale(double sale) { this.tSale = sale;}
262 }
263
```