

```

1 package Ejercicio02;
2
3 public class ColaArrayInt implements ColasInt {
4     //variables
5     private static boolean flag = false; //para controlar la impresion de la linea
6
7     //variables miembro
8     private static final int LONGITUD_POR_DEFECTO = 10;
9     private int maxLongitud; //Tamaño por defecto
10    private int cabeza; //indice del primero de la cola
11    private int fin; //indice del ultimo de la cola
12    private int[] datos; //array que almacena elementos
13
14    //constructores
15    public ColaArrayInt() { this(LONGITUD_POR_DEFECTO); }
16
17    public ColaArrayInt(int max) {
18        this.maxLongitud = max + 1; //un espacio extra
19        this.fin = 0;
20        this.cabeza = 1;
21        datos = new int[maxLongitud];
22    }
23
24    //getters
25    public int getFin() {return this.fin;}
26    public int getCabeza() {return this.cabeza;}
27    public int getMaxLongitud() {return maxLongitud;}
28
29    //metodos
30    public void vaciar() {
31        this.fin = 0;
32        this.cabeza = 1;
33        flag = false;
34    }
35
36    /** Añadir a la cola e */
37    public boolean encolar(int e) {
38        //si no se cumple esta condicion, la cola esta llena
39        if ((this.fin + 2) % this.maxLongitud != this.cabeza){
40            this.fin = (this.fin + 1) % this.maxLongitud; // Incremento circular
41            this.datos[this.fin] = e;
42            flag = true;
43            return true;
44        }
45        return false;
46    }
47
48    /** Eliminar y devolver el primer elemento (cabeza) */
49    public int desencolar() {
50        //si no se cumple, la cola esta vacia
51        if (this.longitud() != 0){

```

```

52         int e = this.datos[this.cabeza];
53         this.cabeza = (this.cabeza + 1) % this.maxLongitud; // Incremento Circular
54         return e;
55     }
56     return 0;
57 }
58
59 /** @return primer valor */
60 public int primero() {
61     //sino se cumple: "La cola está vacía";
62     if(this.longitud() != 0) return this.datos[this.cabeza];
63     return 0;
64 }
65
66 /** @return Cantidad de elementos en la cola */
67 public int longitud(){
68     return ((this.fin + this.maxLongitud) - this.cabeza + 1) % this.maxLongitud;
69 }
70
71 @Override
72 public String toString(){
73
74     String cadena="";
75
76     if (flag == true){
77
78         //mientras la posicion de cabeza no sea mayor que la posicion del fin de la cola
79         if (cabeza <= fin){
80             for(int i = cabeza; i <= fin; i++){
81                 cadena = cadena + datos[i] + " ";
82             }
83         }
84
85         //como es una cola circular se puede dar el caso de que el fin de la cola
86         //sea una posicion menor que el inicio de la cola
87         //debemos leer desde el inicio de la cola hasta el final del array
88         //y posteriormente desde el incicio del array hasta el fin de la colas
89         else {
90             for(int i = cabeza; i < maxLongitud; i++){
91                 cadena = cadena + datos[i] + " ";
92             }
93             for (int i = 0; i <= fin; i++){
94                 cadena = cadena + datos[i] + " ";
95             }
96         }
97     }
98     return cadena;
99
100 }
101
102

```

103 }
104