

```
1 package org.example;
2
3 import java.io.DataInputStream;
4 import java.io.DataOutputStream;
5 import java.io.IOException;
6 import java.net.ServerSocket;
7 import java.net.Socket;
8 import java.util.logging.Level;
9 import java.util.logging.Logger;
10 import java.util.List;
11 import com.google.gson.Gson;
12 import java.lang.reflect.Type;
13 import com.google.gson.reflect.TypeToken;
14 //
15 import javax.print.*;
16 import javax.print.attribute.*;
17 import javax.print.attribute.standard.*;
18 //
19 import java.util.Random;
20 import java.text.SimpleDateFormat;
21 import java.util.Date;
22
23
24 public class servidor {
25
26     public static void main(String[] args) {
27         Random rand = new Random();
28         int numeroTicket = rand.nextInt(900000) +
29         100000;
30         Date fechaActual = new Date();
31         SimpleDateFormat formatoHora = new
32         SimpleDateFormat("HH:mm:ss");
33         String horaActual = formatoHora.format(
34         fechaActual);
35
36         ServerSocket servidor = null;
37         Socket sc = null;
38         int PUERTO = 5000;
39         DataInputStream in;
40         DataOutputStream out;
41         String eleccionesJson;
```

```

39
40     try {
41         servidor = new ServerSocket(PUERTO);
42         System.out.println("SERVIDOR INICIADO");
43
44         while(true)
45         {
46             sc = servidor.accept();
47             System.out.println("SU PEDIDO ES:");
48             in =new DataInputStream(sc.
getInputStream());
49             out =new DataOutputStream(sc.
getOutputStream());
50
51             int numeroMesa = in.readInt();
52             System.out.println("Número de mesa
recibido: " + numeroMesa);
53
54
55             eleccionesJson = in.readUTF();
56             Type listType = new TypeToken<List<
String>>(){}.getType();
57             List<String> elecciones = new Gson().
fromJson(eleccionesJson, listType);
58             for (String eleccion : elecciones) {
59                 System.out.println(eleccion);
60             }
61
62             String mensaje = in.readUTF();//
recepta el total del pedido
63             System.out.println("el total es: "+
mensaje);
64
65
66             String ticket =
67             "-----\n" +
68             "          BIENVENIDO A LEO&IBRA\n"
69             " +
68             "-----\n" +
69             "          TICKET DE PEDIDO #"

```

```

69 + numeroTicket + "\n" +
70
71 "-----\n" +
72 "Fecha y hora de impresión
73 : " + horaActual + "\n" +
74
75 "-----\n" +
76 "Mesa: " + numeroMesa + "\n"
77 +
78
79 "-----\n" +
80 "Elecciones:\n";
81
82 // Agregar cada elección en una nueva línea
83
84 for (String eleccion : elecciones) {
85     eleccion = eleccion.replace("á"
86     , "\u00E1");
87     eleccion = eleccion.replace("é"
88     , "\u00E9");
89     eleccion = eleccion.replace("í"
90     , "\u00ED");
91     eleccion = eleccion.replace("ó"
92     , "\u00F3");
93     eleccion = eleccion.replace("ú"
94     , "\u00FA");
95     eleccion = eleccion.replace("ñ"
96     , "\u00F1");
97
98     ticket += eleccion + "\n"; //
99     Imprime todos los pedidos
100 }
101
102 ticket +=
103 "-----\n";
104 ticket += String.format("Total: $%.
105 2f\n", Double.parseDouble(mensaje));
106 ticket +=
107 "-----\n";
108
109 // Crear el documento para imprimir

```

```

95         DocFlavor flavor = DocFlavor.
        BYTE_ARRAY.AUTOSENSE;
96         Doc doc = new SimpleDoc(ticket.
        getBytes(), flavor, null);
97
98 // Configurar la impresión
99         PrintService service =
        PrintServiceLookup.lookupDefaultPrintService();
100        DocPrintJob job = service.
        createPrintJob();
101        PrintRequestAttributeSet attrs = new
        HashPrintRequestAttributeSet();
102        attrs.add(new Copies(1)); // Número
        de copias
103
104 // Imprimir el documento
105        try {
106            job.print(doc, attrs);
107        } catch (PrintException e) {
108            e.printStackTrace();
109        }
110
111        out.writeUTF("SU PEDIDO A SIDO
        RECIBIDO");
112        System.out.println(
        "*****");
113
114
115        sc.close();
116    }
117
118    }
119    catch (IOException ex)
120    {
121        Logger.getLogger(servidor.class.getName
        ()).log(Level.SEVERE, null, ex);
122    }
123
124 }
125
126 }

```