```java
/**
 * @ (#) Cliente.java
 *
 * Clase Cliente.
 * Es una subclase que hereda de la clase madre Usuario.
 * Cliente es el objeto sobre el que se van a realizar los trabajos en GestionTaller.
 * Representa al dueño de uno o varios vehículos, se incluirán en su ArrayList vehiculos.
 *
 * @author Yassine Marroun
 * @version 1.00 2017/05/24
 */
import java.util.ArrayList;

public class Cliente extends Usuario{

    private String dni;
    private String correo;
    private ArrayList<Vehiculo> vehiculos;
    private ArrayList<Oferta> ofertas;


    public Cliente () {

    }

    public Cliente (String nombre, Integer telefono, String dni, String correo) {
        super(nombre, telefono);
        this.dni = dni;
        this.correo = correo;
        this.vehiculos = new ArrayList<Vehiculo>();
    }


    public String getDni() {
        return dni;
    }

    public void setDni(String dni) {
        this.dni = dni;
    }

    public String getCorreo() {
        return correo;
    }

    public void setCorreo(String correo) {
        this.correo = correo;
    }

    public ArrayList<Vehiculo> getVehiculos() {
        return vehiculos;
    }

    public void setVehiculos(ArrayList<Vehiculo> vehiculos) {
        this.vehiculos = vehiculos;
    }

    public ArrayList<Oferta> getOfertas() {
        return ofertas;
    }

    public void setOfertas(ArrayList<Oferta> ofertas) {
```

```
63          this.ofertas = ofertas;
64      }
65
66
67      // El método addVehiculo añade un vehiculo a la lista de vehículos de un cliente.
68
69      public void addVehiculo(Vehiculo vehiculo) {
70
71          this.vehiculos.add(vehiculo);
72      }
73
74
75      // Con el método nuevoCliente, obtenemos los datos de un cliente que acude al taller.
76
77      public Cliente nuevoCliente() {
78              System.out.print("Introduzca el nombre del Cliente: ");
79              String nombre = Menu.sc.nextLine();
80              System.out.print("Introduzca el Telefono: ");
81              String tfno = Menu.sc.nextLine();
82              Integer telefono = Integer.parseInt(tfno);
83              System.out.print("Introduzca el DNI: ");
84              String dni = Menu.sc.nextLine();
85              System.out.print("Introduzca el correo electronico: ");
86              String correo = Menu.sc.nextLine();
87              Cliente cliente = new Cliente(nombre, telefono, dni, correo);
88              return cliente;
89      }
90
91
92      // El método enviarOfertas imprime las ofertas que le corresponden a un cliente.
93      // Tal como se comenta en el enunciado de la práctica, es una simulación de enviar las
    ofertas a un cliente.
94
95      public String enviarOfertas(){
96         String enviar = "";
97         enviar = "El cliente " + this.nombre + " con correo " + this.correo;
98         if (this.ofertas==null){
99             enviar = enviar + " no tiene ofertas: \n";
100        } else{
101            enviar = enviar + " tiene las ofertas: \n";
102            for (Oferta ofe: this.ofertas){
103                enviar = enviar + ofe.toString();
104            }
105        }
106        return enviar;
107     }
108
109
110     @Override
111     public String toString() {
112         String datosCliente = super.toString() +
113                               "\n DNI: " + dni +
114                                "\n eMail: " + correo +
115                                "\n Vehiculos: " + vehiculos.toString() +
116                                "\n tiene Ofertas: ";
117         if (ofertas==null || ofertas.size()<=0){
118             datosCliente = datosCliente + "no";
119         } else {
120             datosCliente = datosCliente + "si";
121         }
122         datosCliente = datosCliente + "\n ********** \n ";
123         return datosCliente;
```

```
124     }
125 }
```