



Universidad Fidélitas.

Proyecto de Introducción a la Programación SC-
202 KT2pm-5pm.

Profesor:
Esteban Marín Chinchilla.

Integrantes:
Mathiun Zamora Rodríguez.
Rogelio Sánchez.
Esteban Ugalde Rojas.




Introducción.

Elegimos basar nuestro proyecto en una pulpería o un minisúper, ya que es algo muy común en nuestras comunidades y se nos hacia muy interesante crear un programa para estas.

Este proyecto lo queremos enfocar para los trabajadores, para así crear un ambiente de trabajo más fácil para ellos a la hora de realizar varias funciones que los clientes necesitan, ya sea realizar un pago, validar un cupón o un descuento etc.

Nosotros decidimos nombrar nuestro proyecto como “Mini Super Fidélitas”.

A continuación se mostraran las clases, atributos, métodos llevará nuestro proyecto y el diagrama de cada uno de estos.






Objetivo General y Objetivos Específicos.

Objetivo General:

- Basar nuestro proyecto en una pulpería o un minisúper, ya que es algo muy común en nuestras comunidades y se nos hacia muy interesante crear un programa para estas.

Objetivos Específicos:

- Enfocar este proyecto para los trabajadores, para así crear un ambiente de trabajo más fácil para ellos a la hora de realizar varias funciones que los clientes necesitan, ya sea realizar un pago, validar un cupón o un descuento etc.
 - Decidir nombrar nuestro proyecto como “Mini Super Fidélitas”.
- 

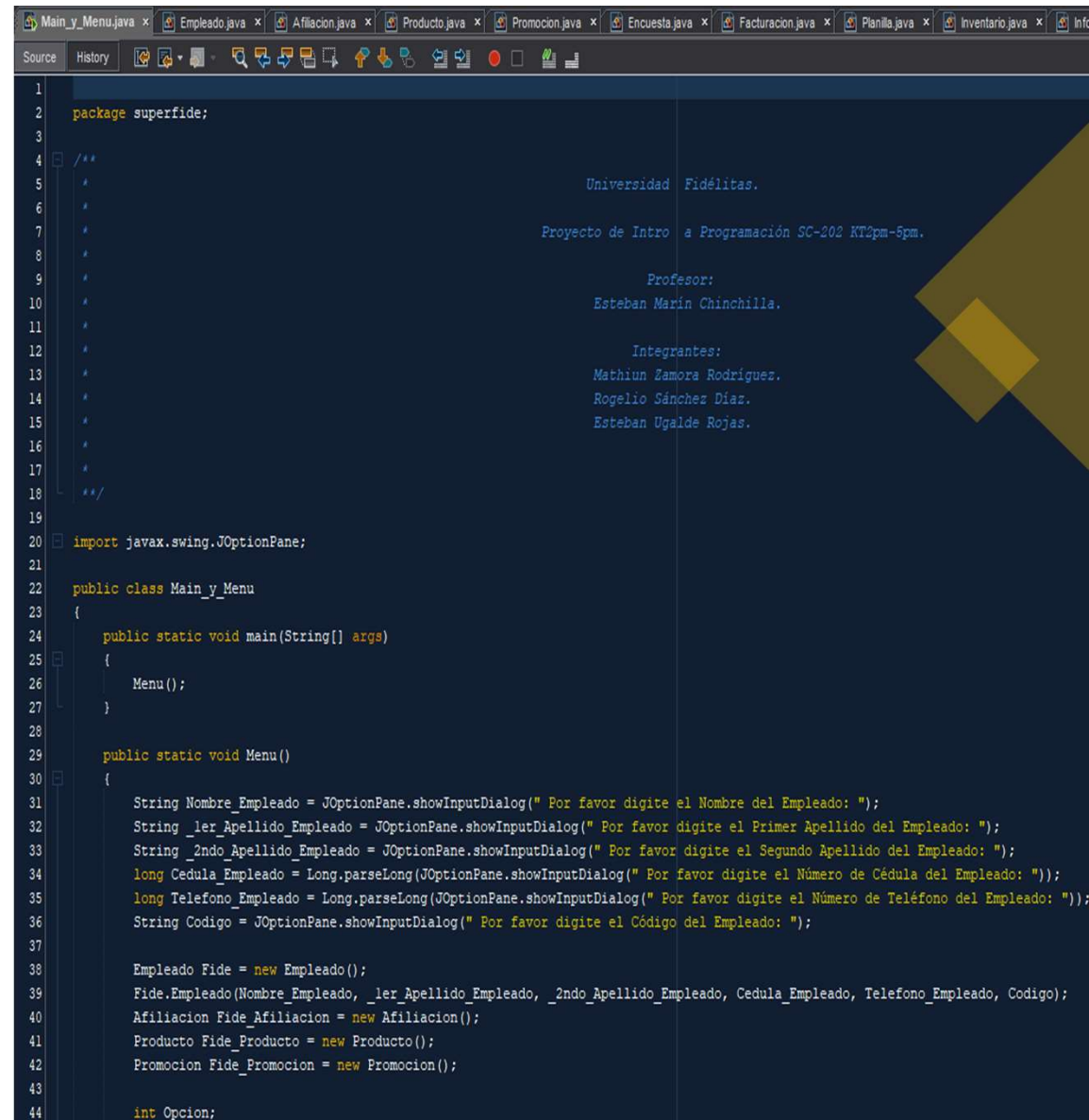
Clase Main.

En esta clase principal se inicia el programa.

Esta clase solicita los datos del empleado (Nombre, 1er Apellido, 2do Apellido, Cédula, Código, Teléfono) que está atendiendo al cliente para poder ingresar al menú del cajero y atención al cliente.

Cuando el empleado ya digito los datos se le despliega la ventana del menú para poder escoger una opción (1) Afiliación, 2) Compra de Productos, 3) Validación del Cupón de Descuento, 4) Encuesta, 5) Facturación, 6) Planilla, 7) Inventario de Productos, 0) Salida.)

Cada opción tiene un diferente número para poder hacer los procesos correspondientes mediante las respectivas clases de cada una de las opciones. Al escoger el número al empleado lo lleva a clase respectiva y en esta se hacen los procesos correspondientes. En caso de seleccionar la opción 0 de salida, el empleado se sale del menú y se finaliza del programa, igualmente si yo escogí una clase para hacer los procesos correspondientes y ya termino se termina el programa automáticamente sin necesidad de escoger la opción de salida.



```
1 package superfide;
2
3
4 /**
5  * Universidad Fidéltas.
6  *
7  * Proyecto de Intro a Programación SC-202 KT2pm-5pm.
8  *
9  * Profesor:
10  * Esteban Marín Chinchilla.
11  *
12  * Integrantes:
13  * Mathiun Zamora Rodriguez.
14  * Rogelio Sánchez Díaz.
15  * Esteban Ugalde Rojas.
16  *
17  */
18
19
20 import javax.swing.JOptionPane;
21
22 public class Main_y_Menu
23 {
24     public static void main(String[] args)
25     {
26         Menu();
27     }
28
29     public static void Menu()
30     {
31         String Nombre_Empleado = JOptionPane.showInputDialog(" Por favor digite el Nombre del Empleado: ");
32         String _1er_Apellido_Empleado = JOptionPane.showInputDialog(" Por favor digite el Primer Apellido del Empleado: ");
33         String _2ndo_Apellido_Empleado = JOptionPane.showInputDialog(" Por favor digite el Segundo Apellido del Empleado: ");
34         long Cedula_Empleado = Long.parseLong(JOptionPane.showInputDialog(" Por favor digite el Número de Cédula del Empleado: "));
35         long Telefono_Empleado = Long.parseLong(JOptionPane.showInputDialog(" Por favor digite el Número de Teléfono del Empleado: "));
36         String Codigo = JOptionPane.showInputDialog(" Por favor digite el Código del Empleado: ");
37
38         Empleado Fide = new Empleado();
39         Fide.Empleado(Nombre_Empleado, _1er_Apellido_Empleado, _2ndo_Apellido_Empleado, Cedula_Empleado, Telefono_Empleado, Codigo);
40         Afiliacion Fide_Afiliacion = new Afiliacion();
41         Producto Fide_Producto = new Producto();
42         Promocion Fide_Promocion = new Promocion();
43
44         int Opcion;
```

Clase Main.

```
92
93
94     case 4:
95         Encuesta Form = new Encuesta();
96         Form.getPregunta_1();
97         Form.getPregunta_2();
98         Form.getPregunta_3();
99         // Code Block
100     break;
101
102     case 5:
103         Facturacion Fide_Fact = new Facturacion(Fide, Fide_Afiliacion, Fide_Producto, Fide_Promocion);
104         Fide_Fact.Facturacion();
105         // Code Block
106     break;
107
108     case 6:
109         Planilla Fide_Planilla = new Planilla();
110         Fide_Planilla.Planilla();
111         // Code Block
112     break;
113
114     case 7:
115         Inventario Fide_Inv = new Inventario();
116         Fide_Inv.Inventario();
117         // Code Block
118     break;
119
120     case 0:
121         // Code Block
122     break;
123
124     default:
125         JOptionPane.showMessageDialog(null, " Opción incorrecta. ");
126
127         System.out.println(" ");
128         System.out.println(" Opción incorrecta. ");
129         System.out.println(" ");
130
131 }
132
133 while (Opcion!=0);
134 }
```

```
46
47
48     do
49     {
50         Opcion = Integer.parseInt(JOptionPane.showInputDialog(null, " \n "
51         + " Bienvenid@: " +Nombre_Empleado+ " " + _ler_Apellido_Empleado+ " "
52         + " \n "
53         + " Atención a los Clientes: "
54         + " \n Digite el número según la opción que desee: "
55         + " \n 1) Afiliación del Cliente. "
56         + " \n 2) Compra de Productos. "
57         + " \n 3) Validación del Cupón de Descuento. "
58         + " \n 4) Encuesta. "
59         + " \n 5) Facturación. \n "
60         + " \n "
61         + " Atención a los Encargados de Planilla: "
62         + " \n Digite el número según la opción que desee: "
63         + " \n 6) Planilla. \n "
64         + " \n "
65         + " Atención a los Empleados: "
66         + " \n Digite el número según la opción que desee: "
67         + " \n 7) Inventario de Productos. "
68         + " \n 0) Salida. \n "
69         + " \n"));
70
71         switch (Opcion)
72         {
73             case 1:
74                 String Nombre_Cliente = JOptionPane.showInputDialog(" Por favor digite el Nombre del Cliente: ");
75                 String _ler_Apellido_Cliente = JOptionPane.showInputDialog(" Por favor digite el Primer Apellido del Cliente: ");
76                 String _2ndo_Apellido_Cliente = JOptionPane.showInputDialog(" Por favor digite el Segundo Apellido del Cliente: ");
77                 long Cedula_Cliente = Long.parseLong(JOptionPane.showInputDialog(" Por favor digite el Número de Cédula del Cliente: "));
78                 String Telefono_Cliente = Long.parseLong(JOptionPane.showInputDialog(" Por favor digite el Número de Teléfono del Cliente: "));
79                 String Correo_Cliente = JOptionPane.showInputDialog(" Por favor digite el Correo Electrónico del Cliente: ");
80                 String Direccion_Cliente = JOptionPane.showInputDialog(" Por favor digite la Direccion del Cliente: ");
81                 Fide_Afiliacion.Afiliacion(Nombre_Cliente, _ler_Apellido_Cliente, _2ndo_Apellido_Cliente, Cedula_Cliente, Telefono_Cliente, Direccion_Cliente);
82                 // Code Block
83             break;
84
85             case 2:
86                 Fide_Producto.Llenar();
87                 // Code Block
88             break;
89
90             case 3:
91                 Fide_Promocion.Precio_Promocion(Fide_Producto);
92                 // Code Block
93             break;
94
95             // ... (other cases) ...
96         }
97     }
98 }
```

Input

Bienvenid@: Esteban Ugalde Rojas, Cédula: 118360459, Teléfono: 84199675, Código: 179JeampiEUR101112 al programa para la atención a los Clientes, Empleados y Encargados de Planilla.

Atención a los Clientes:
Digite el número según la opción que desee:
1) Afiliación del Cliente.
2) Compra de Productos.
3) Validación del Cupón de Descuento.
4) Encuesta.
5) Facturación.

Atención a los Encargados de Planilla:
Digite el número según la opción que desee:
6) Planilla.

Atención a los Empleados:
Digite el número según la opción que desee:
7) Inventario de Productos.
0) Salida.

OK Cancel

Clase "Empleado".

En esta clase se almacenan los datos del empleado al ser ingresados en la clase main antes de ingresar a la ventana del menú, para poder imprimir los datos del empleado a la de hacer la facturación escogiendo la opción 5 en el menú del main y la clase que hace todo el proceso de la impresión de la factura es la clase factura.

```
package superfide;
import javax.swing.JOptionPane;

public class Empleado
{
    private String Nombre_Empleado;
    private String _1er_Apellido_Empleado;
    private String _2do_Apellido_Empleado;
    private StringCodigo_Empleado;
    private long Cedula_Empleado;
    private long Telefono_Empleado;

    public void Empleado(String Nombre_E, String _1er_Apellido_E, String _2do_Apellido_E, long Cedula_E, long Telefono_E, String Codigo)
    {
        Nombre_Empleado = Nombre_E;
        _1er_Apellido_Empleado = _1er_Apellido_E;
        _2do_Apellido_Empleado = _2do_Apellido_E;
        Codigo_Empleado = Codigo;
        Cedula_Empleado = Cedula_E;
        Telefono_Empleado = Telefono_E;
    }

    public void Imprimir_Empleado()
    {
        JOptionPane.showMessageDialog(null, "\n "
            + "Nombre: " + Nombre_Empleado + " " + _1er_Apellido_Empleado + " " + _2do_Apellido_Empleado + " \n "
            + "Codigo: " + Codigo_Empleado + " \n "
            + "Cedula: " + Cedula_Empleado + " \n "
            + "Telefono: " + Telefono_Empleado + " \n "
            + " \n ");

        System.out.println("\n ");
        System.out.println("Nombre: " + Nombre_Empleado + " " + _1er_Apellido_Empleado + " " + _2do_Apellido_Empleado + " \n ");
        System.out.println("Codigo: " + Codigo_Empleado + " \n ");
        System.out.println("Cedula: " + Cedula_Empleado + " \n ");
        System.out.println("Telefono: " + Telefono_Empleado + " \n ");
    }

    public String getNombre_Empleado()
    {
        return Nombre_Empleado;
    }

    public void setNombre_Empleado(String Nombre_Empleado)
    {
        this.Nombre_Empleado = Nombre_Empleado;
    }

    public String get1er_Apellido_Empleado()
    {
        return _1er_Apellido_Empleado;
    }

    public void set1er_Apellido_Empleado(String _1er_Apellido_Empleado)
    {
        this._1er_Apellido_Empleado = _1er_Apellido_Empleado;
    }

    public String get2do_Apellido_Empleado()
    {
        return _2do_Apellido_Empleado;
    }

    public void set2do_Apellido_Empleado(String _2do_Apellido_Empleado)
    {
        this._2do_Apellido_Empleado = _2do_Apellido_Empleado;
    }

    public String getCodigo_Empleado()
    {
        return Codigo_Empleado;
    }

    public void setCodigo_Empleado(String Codigo_Empleado)
    {
        this.Codigo_Empleado = Codigo_Empleado;
    }

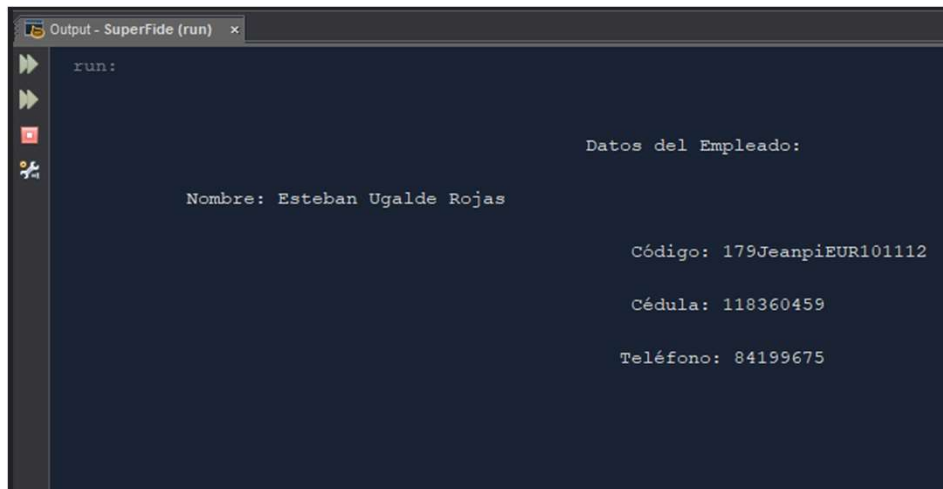
    public long getCedula_Empleado()
    {
        return Cedula_Empleado;
    }

    public void setCedula_Empleado(long Cedula_Empleado)
    {
        this.Cedula_Empleado = Cedula_Empleado;
    }

    public long getTelefono_Empleado()
    {
        return Telefono_Empleado;
    }

    public void setTelefono_Empleado(long Telefono_Empleado)
    {
        this.Telefono_Empleado = Telefono_Empleado;
    }
}
```

Clase “Empleado”.



```
run:

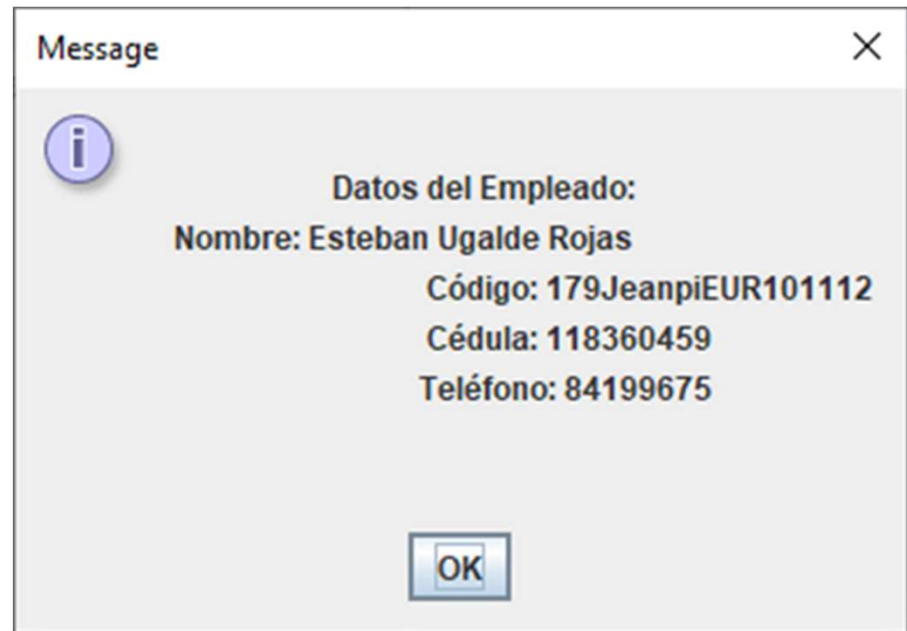
Datos del Empleado:

Nombre: Esteban Ugalde Rojas

Codigo: 179JeanpiEUR101112

Cedula: 118360459

Telefono: 84199675
```



Clase “Afiliación”.

En esta clase el empleado procede a afiliarse al cliente si el cliente le solicita al empleado que escoja la opción 1 para realizar la afiliación del cliente.

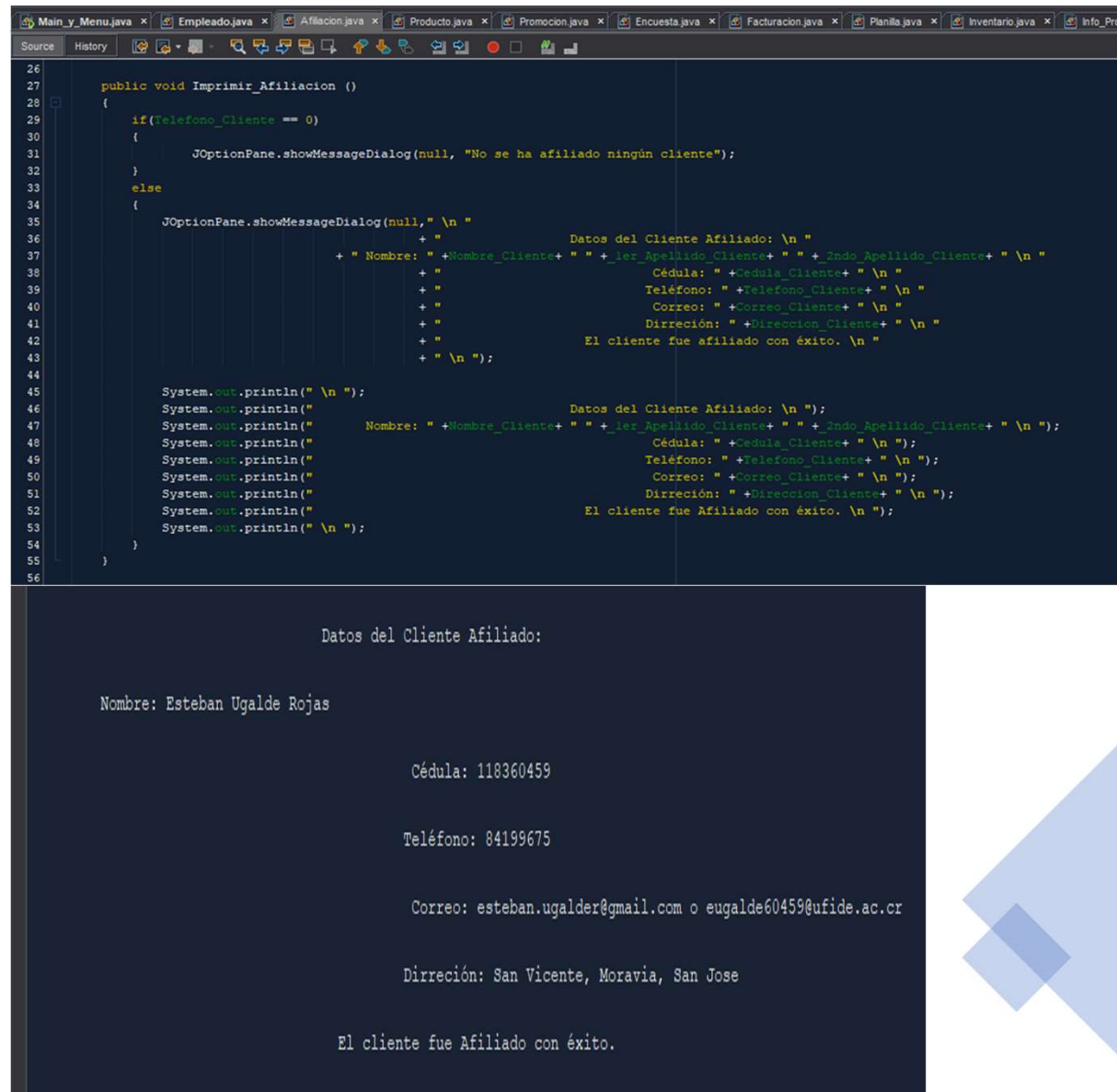
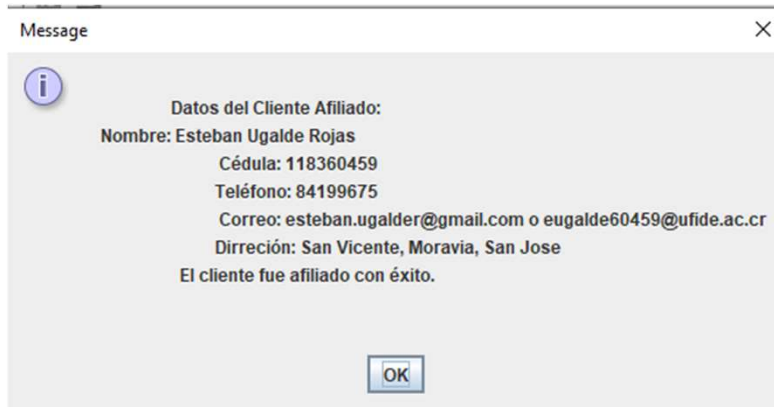
Después el empleado digita los datos del funcionario en la clase main y se almacenan en la clase “Afiliación” para poder imprimir en la factura, la afiliación y todo lo demás correspondiente a la factura a la hora de hacer la facturación mediante la clase “Facturación”.

En caso de que el cliente no quiera afiliarse y el encargado termine con el proceso de registro de productos comprados registrados en la clase “Producto”, se procede a imprimir la factura mediante la clase Facturación con lo siguiente:

- Los datos del empleado almacenado en la clase “Empleado” al haber sido registrado en la clase principal antes del menú.
- Los datos de los productos registrados en la clase “Producto”.
- Y en caso de que el cliente le solicite al cajero seleccionar la opción 3 para aplicar un descuento mediante la clase “Promoción”, se procede a imprimir en la factura los datos de los productos registrados en la clase “Producto” antes de aplicar el descuento y también se imprime los precio con el descuento aplicado.

```
1 package superfide;
2
3
4 import javax.swing.JOptionPane;
5
6 public class Afiliacion
7 {
8     private String Nombre_Cliente;
9     private String _1er_Apellido_Cliente;
10    private String _2ndo_Apellido_Cliente;
11    private long Cedula_Cliente;
12    private long Telefono_Cliente = 0;
13    private String Correo_Cliente;
14    private String Direccion_Cliente;
15
16    public void Afiliacion(String Nombre_C, String _1er_Apellido_C, String _2ndo_Apellido_C, long Cedula_C, long Telefono_C, String Correo, String Direccion)
17    {
18        Nombre_Cliente = Nombre_C;
19        _1er_Apellido_Cliente = _1er_Apellido_C;
20        _2ndo_Apellido_Cliente = _2ndo_Apellido_C;
21        Cedula_Cliente = Cedula_C;
22        Telefono_Cliente = Telefono_C;
23        Correo_Cliente = Correo;
24        Direccion_Cliente = Direccion;
25    }
26
27
28    public String getNombre_Cliente()
29    {
30        return Nombre_Cliente;
31    }
32
33    public void setNombre_Cliente(String Nombre_Cliente)
34    {
35        this.Nombre_Cliente = Nombre_Cliente;
36    }
37
38    public long getCedula_Cliente()
39    {
40        return Cedula_Cliente;
41    }
42
43    public void setCedula_Cliente(long Cedula_Cliente)
44    {
45        this.Cedula_Cliente = Cedula_Cliente;
46    }
47
48    public String get1er_Apellido_Cliente()
49    {
50        return _1er_Apellido_Cliente;
51    }
52
53    public void set1er_Apellido_Cliente(String _1er_Apellido_Cliente)
54    {
55        this._1er_Apellido_Cliente = _1er_Apellido_Cliente;
56    }
57
58    public String get2ndo_Apellido_Cliente()
59    {
60        return _2ndo_Apellido_Cliente;
61    }
62
63    public void set2ndo_Apellido_Cliente(String _2ndo_Apellido_Cliente)
64    {
65        this._2ndo_Apellido_Cliente = _2ndo_Apellido_Cliente;
66    }
67
68
69    public long getTelefono_Cliente()
70    {
71        return Telefono_Cliente;
72    }
73
74    public void setTelefono_Cliente(long Telefono_Cliente)
75    {
76        this.Telefono_Cliente = Telefono_Cliente;
77    }
78
79    public String getCorreo_Cliente()
80    {
81        return Correo_Cliente;
82    }
83
84    public void setCorreo_Cliente(String Correo_Cliente)
85    {
86        this.Corrreo_Cliente = Correo_Cliente;
87    }
88
89    public String getDireccion_Cliente()
90    {
91        return Direccion_Cliente;
92    }
93
94    public void setDireccion_Cliente(String Direccion_Cliente)
95    {
96        this.Direccion_Cliente = Direccion_Cliente;
97    }
98
99 }
```

Clase “Afiliación”.



En esta clase el empleado ingresa los datos de los producto después de que el empleado seleccione la opción 2.

Por medio de esta clase y la clase de facturación y en caso de aplicar el descuento mediante la clase "Promoción" se imprimen lo siguiente:

- Los datos del empleado almacenado en la clase "Empleado" al haber sido registrado en la clase principal antes del menú.
- Los datos de los productos registrados en la clase "Producto".
- Y en caso de que el cliente le solicite al cajero seleccionar la opción 3 para aplicar un descuento mediante la clase "Promoción", se procede a imprimir en la factura los datos de los productos registrados en la clase "Producto" antes de aplicar el descuento y también se imprime los precio con el descuento aplicado.
- Y en caso de que el cliente le solicite al cajero seleccionar la opción 1 para afiliarse al cliente mediante la clase "Afiliación", se procede a imprimir en la factura los datos de del cliente digitados en la clase principal y almacenados en la clase "Afiliación".

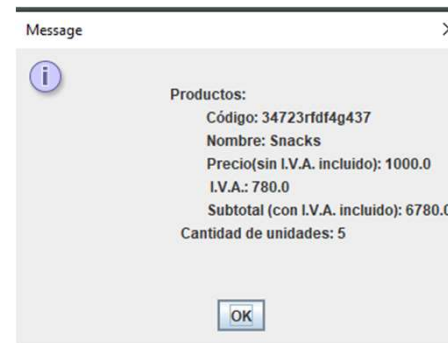
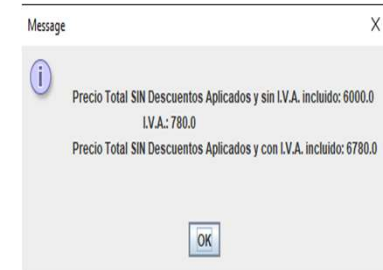
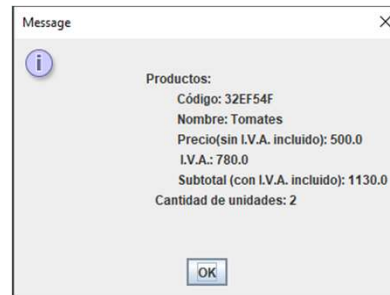
```
148
149     public long getCantidad_Producto()
150     {
151         return Cantidad_Producto;
152     }
153
154     public void setCantidad_Producto(long Cantidad_Producto)
155     {
156         this.Cantidad_Producto = Cantidad_Producto;
157     }
158 }
159
160
```

[illegible]

```
Main_y_Menu.java x Empleado.java x Afiliacion.java x Producto.java x Promocion.java x Encuesta.java x
Source History
107 else
108 {
109     JOptionPane.showMessageDialog(null, "\n "
110                                     + " No hay productos para mostrar. \n "
111                                     + " \n ");
112
113     System.out.println("\n ");
114     System.out.println(" No hay productos para mostrar. \n");
115     System.out.println("\n ");
116 }
117
118
119 public String getCodigo_Producto()
120 {
121     return Codigo_Producto;
122 }
123
124 public void setCodigo_Producto(String Codigo_Producto)
125 {
126     this.Codigo_Producto = Codigo_Producto;
127 }
128
129 public String getNombre_Producto()
130 {
131     return Nombre_Producto;
132 }
133
134 public void setNombre_Producto(String Nombre_Producto)
135 {
136     this.Nombre_Producto = Nombre_Producto;
137 }
138
139 public double getPrecio_Producto()
140 {
141     return Precio_Producto;
142 }
143
144 public void setPrecio_Producto(double Precio_Producto)
145 {
146     this.Precio_Producto = Precio_Producto;
147 }
148
```

```
Main_y_Menu.java x | Empleado.java x | Afiliacion.java x | Producto.java x | Promocion.java x | Escencia.java x  
Source History [Icons] [Icons]  
1 package superfide;  
2  
3  
4 import javax.swing.JOptionPane;  
5  
6 public class Producto  
7 {  
8     private String Codigo[];  
9     private String Nombre[];  
10    private double Precio[];  
11    private double Subtotal[];  
12    private long Cantidad[];  
13  
14    private String Codigo_Producto = "";  
15    private String Nombre_Producto = "";  
16    private double Precio_Producto = 0;  
17    private double IVA = 0;  
18    private double SUBT = 0;  
19    private long Cantidad_Producto = 0;  
20  
21    public void Producto(String Nombre, String Codigo, double Precio, long Cantidad)  
22    {  
23        Nombre_Producto = Nombre;  
24        Codigo_Producto = Codigo;  
25        Precio_Producto = Precio;  
26        Cantidad_Producto = Cantidad;  
27    }  
28
```

Clase "Producto".



```
Output - SuperFide (run) x
1      Productos:
      Código: 32EF54F
      Nombre: Tomates
      Precio(sin I.V.A. incluido): 500.0
      I.V.A.: 780.0
      Subtotal (con I.V.A. incluido): 1130.0
      Cantidad de unidades: 2

2      Productos:
      Código: 34723rfd4g437
      Nombre: Snacks
      Precio(sin I.V.A. incluido): 1000.0
      I.V.A.: 780.0
      Subtotal (con I.V.A. incluido): 6780.0
      Cantidad de unidades: 5
```

```
Precio Total SIN Descuentos Aplicados y sin I.V.A. incluido: 6000.0
I.V.A.: 780.0
Precio Total SIN Descuentos Aplicados y con I.V.A. incluido: 6780.0
```


Clase “Promoción”.

En caso de que el cliente le solicite al cajero seleccionar la opción 3 para aplicar un descuento mediante esta clase, se procede a ingresar el porcentaje solicitado por el cliente para convertirlo en decimal para que el empleado aplique el descuento correctamente.

Después de ingresar el porcentaje solicitado, de la conversión y calculo respectivo se procede a imprimir mediante la clase “Facturación” lo siguiente:

- Los datos del empleado almacenado en la clase “Empleado” al haber sido registrado en la clase principal antes del menú.
- Los datos de los productos registrados en la clase “Producto”.
- Y en caso de que el cliente le solicite al cajero seleccionar la opción 3 para aplicar un descuento mediante la clase “Promoción”, se procede a imprimir en la factura los datos de los productos registrados en la clase “Producto” antes de aplicar el descuento y también se imprime los precio con el descuento aplicado.
- Y en caso de que el cliente le solicite al cajero seleccionar la opción 1 para afiliarse al cliente mediante la clase “Afiliación”, se procede a imprimir en la factura los datos de del cliente digitados en la clase principal y almacenados en la clase “Afiliación”.

```
1 package superfide;
2
3 import javax.swing.JOptionPane;
4
5 public class Promocion
6 {
7     private double Cupon;
8     private double Precio_Cupon;
9
10     Producto Fide_Producto = new Producto();
11
12     public void Precio_Promocion(Producto P)
13     {
14         // Se le pide al usuario el porcentaje con numeros enteros para despues convertirlo en decimal(double / float)
15         Cupon = Double.parseDouble(JOptionPane.showInputDialog(" Digite el porcentaje del cupon de descuento a canjear: "));
16         Cupon = Cupon / 100;
17         Precio_Cupon = P.Operation() - (P.Operation() * Cupon);
18     }
19
20     public void Imprimir_Promocion()
21     {
22         if (Cupon != 0)
23         {
24             JOptionPane.showMessageDialog(null, "\n "
25                 + " El porcentaje del cupón del descuento de la promoción es: " +Cupon+ " \n "
26                 + " El Precio Total CON el Descuento del Cupón Aplicado (con I.V.A. incluido) es de: " +Precio_Cupon+ " \n "
27                 + " \n ");
28
29             System.out.println("\n ");
30             System.out.println(" El porcentaje del cupón del descuento de la promoción es: " +Cupon+ " \n ");
31             System.out.println(" El Precio Total CON la promoción del Descuento del Cupon Aplicado (con I.V.A. incluido) es de: " +Precio_Cupon+ " \n ");
32             System.out.println("\n ");
33         }
34     }
35 }
36
```

```
37 else
38 {
39     JOptionPane.showMessageDialog(null, "\n "
40         + " No se ha aplicado ningún tipo de promoción de descuento de cupón. \n "
41         + " \n ");
42
43     System.out.println("\n ");
44     System.out.println(" No se ha aplicado ningún tipo de promoción de descuento de cupón. \n ");
45     System.out.println("\n ");
46 }
47
48 public double getCupon()
49 {
50     return Cupon;
51 }
52
53 public void setCupon(double Cupon)
54 {
55     this.Cupon = Cupon;
56 }
57
58 public double getPrecio_Cupon()
59 {
60     return Precio_Cupon;
61 }
62
63 public void setPrecio_Cupon(double Precio_Cupon)
64 {
65     this.Precio_Cupon = Precio_Cupon;
66 }
67
68 }
69
```

El porcentaje del cupón del descuento de la promoción es: 0.1

El Precio Total CON la promoción del Descuento del Cupon Aplicado (con I.V.A. incluido) es de: 5400.0

Message

X



El porcentaje del cupón del descuento de la promoción es: 0.1

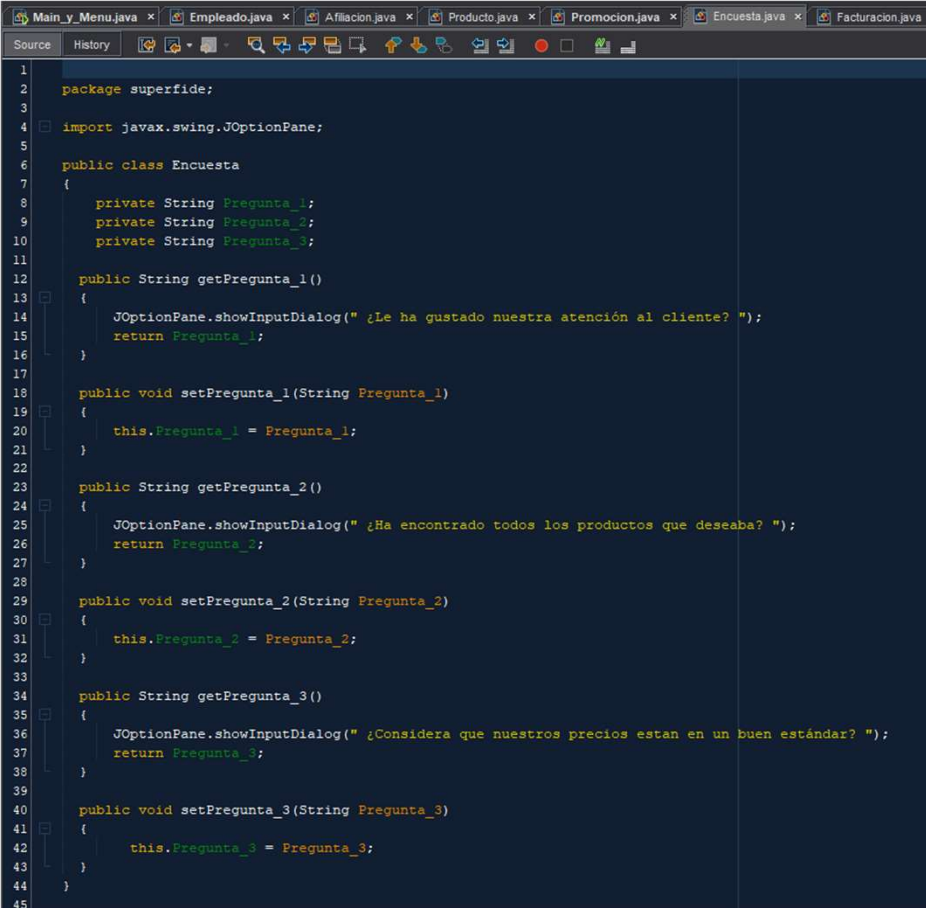
El Precio Total CON el Descuento del Cupón Aplicado (con I.V.A. incluido) es de: 5400.0

OK

Clase “Promoción”.

Clase “Encuesta”.

En esta clase al seleccionar la opción 4 el cliente procede a realizar una encuesta opcional, sobre la atención otorgada al clientes, el estándar de los precios y la disponibilidad de los productos.

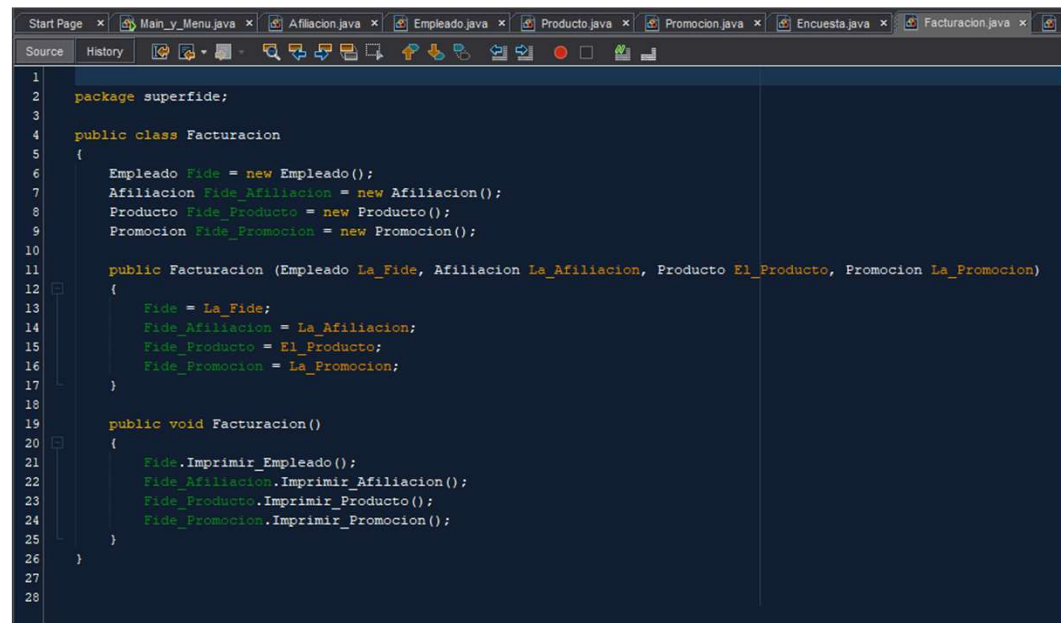
A screenshot of a Java IDE with multiple tabs open: Main_y_Menu.java, Empleado.java, Afiliacion.java, Producto.java, Promocion.java, Encuesta.java, and Facturacion.java. The 'Encuesta.java' tab is active, showing the following code:

```
1
2 package superfide;
3
4 import javax.swing.JOptionPane;
5
6 public class Encuesta
7 {
8     private String Pregunta_1;
9     private String Pregunta_2;
10    private String Pregunta_3;
11
12    public String getPregunta_1()
13    {
14        JOptionPane.showInputDialog(" ¿Le ha gustado nuestra atención al cliente? ");
15        return Pregunta_1;
16    }
17
18    public void setPregunta_1(String Pregunta_1)
19    {
20        this.Pregunta_1 = Pregunta_1;
21    }
22
23    public String getPregunta_2()
24    {
25        JOptionPane.showInputDialog(" ¿Ha encontrado todos los productos que deseaba? ");
26        return Pregunta_2;
27    }
28
29    public void setPregunta_2(String Pregunta_2)
30    {
31        this.Pregunta_2 = Pregunta_2;
32    }
33
34    public String getPregunta_3()
35    {
36        JOptionPane.showInputDialog(" ¿Considera que nuestros precios estan en un buen estándar? ");
37        return Pregunta_3;
38    }
39
40    public void setPregunta_3(String Pregunta_3)
41    {
42        this.Pregunta_3 = Pregunta_3;
43    }
44 }
45
```


Clase “Facturación”.

En esta clase mediante la opción 5 se procede a imprimir la factura con los siguientes:

- Los datos del empleado almacenado en la clase “Empleado” al haber sido registrado en la clase principal antes del menú.
- Los datos de los productos registrados en la clase “Producto”.
- Y en caso de que el cliente le solicite al cajero seleccionar la opción 3 para aplicar un descuento mediante la clase “Promoción”, se procede a imprimir en la factura los datos de los productos registrados en la clase “Producto” antes de aplicar el descuento y también se imprime los precio con el descuento aplicado.
- Y en caso de que el cliente le solicite al cajero seleccionar la opción 1 para afiliarse al cliente mediante la clase “Afiliación”, se procede a imprimir en la factura los datos de del cliente digitados en la clase principal y almacenados en la clase “Afiliación”.



```
1 package superfide;
2
3
4 public class Facturacion
5 {
6     Empleado fide = new Empleado();
7     Afiliacion fide_Afiliacion = new Afiliacion();
8     Producto fide_Producto = new Producto();
9     Promocion fide_Promocion = new Promocion();
10
11     public Facturacion (Empleado La_Fide, Afiliacion La_Afiliacion, Producto El_Producto, Promocion La_Promocion)
12     {
13         fide = La_Fide;
14         fide_Afiliacion = La_Afiliacion;
15         fide_Producto = El_Producto;
16         fide_Promocion = La_Promocion;
17     }
18
19     public void Facturacion()
20     {
21         fide.Imprimir_Empleado();
22         fide_Afiliacion.Imprimir_Afiliacion();
23         fide_Producto.Imprimir_Producto();
24         fide_Promocion.Imprimir_Promocion();
25     }
26 }
27
28
```

Clase “Planilla”.

En esta clase, si el empleado selecciona la opción 6, se procede a digitar los datos, los salarios por hora, horas trabajadas y la zona de trabajo del empleado.

Después de digitar lo mencionado anteriormente se prosigue a calcular las horas extras, deducciones(Caja (CCSS), Banco Popular (BPDC), Sindicato, Ahorro Personal y/o Navideño y Total de deducciones), Comisión por la zona de trabajo del empleado, Salarios Brutos y Salario Neto.

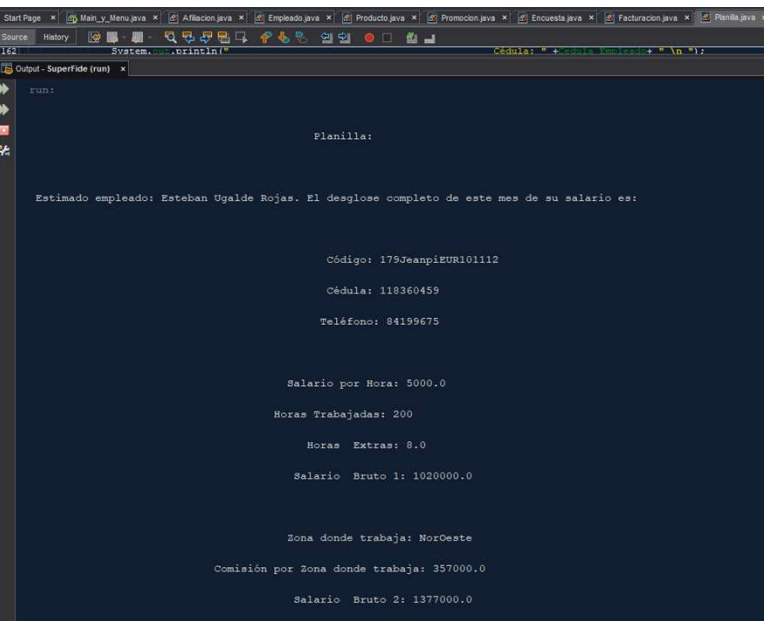
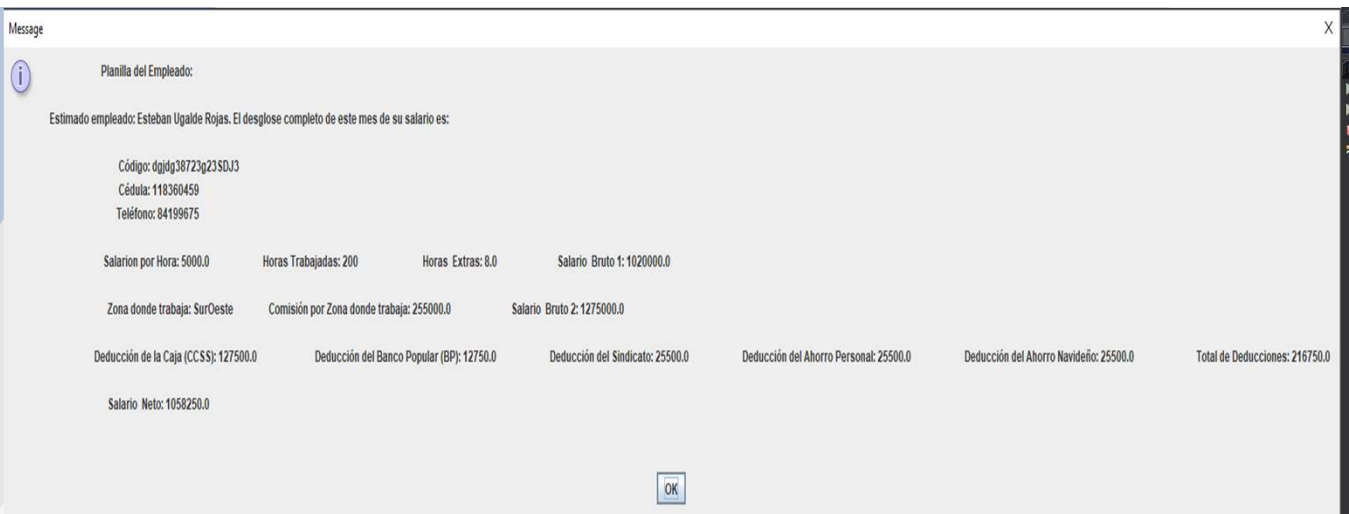
Después de calcular lo mencionado anteriormente se prosigue a imprimir los datos, los salarios por hora, horas trabajadas y la zona de trabajo del empleado las horas extras, deducciones(Caja (CCSS), Banco Popular (BPDC), Sindicato, Ahorro Personal y/o Navideño y Total de deducciones), Comisión por la zona de trabajo del empleado, Salarios Brutos y Salario Neto.

```
Man_y_Menu.java x Afiliacion.java x Empleado.java x Producto.java x Promocion.java x Encuesta.java x Facturacion.java x Planilla.java x
Source History
1 package superfide;
2
3 import javax.swing.JOptionPane;
4
5 public class Planilla
6 {
7     String Nombre_Empleado;
8     String Codigo_Empleado;
9     String _1er_Apellido;
10    String _2do_Apellido;
11    long Cedula_Empleado;
12    long Telefono_Empleado;
13
14    long Horas_Trabajadas;
15    double Horas_Extras;
16    double Salario_por_Hora;
17    double Salario_Bruto_1;
18
19    String Zona;
20    double Comision_por_Zona_donde_trabaja;
21    double Salario_Bruto_2;
22
23    double Deduccion_de_la_Caja_CCSS;
24    double Deduccion_del_Banco_Popular_BPDC;
25    String Respuesta_1;
26    String Respuesta_2;
27    String Respuesta_3;
28    double Deduccion_del_Sindicato;
29    double Deduccion_del_Ahorro_Personal;
30    double Deduccion_del_Ahorro_Navideño;
31    double Total_de_Deducciones;
32
33    double Salario_Neto;
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57 Zona = JOptionPane.showInputDialog(" Digite la Zona donde trabaja el Empleado (Central, Norte, Sur, Este, Oeste, NorEste, SurEste, NorOeste, SurOeste, Otra)");
58 if ("Central".equals(Zona))
59 {
60     Comision_por_Zona_donde_trabaja = Salario_Bruto_1 * 0.05;
61 }
62 else if ("Norte".equals(Zona))
63 {
64     Comision_por_Zona_donde_trabaja = Salario_Bruto_1 * 0.10;
65 }
66 else if ("Sur".equals(Zona))
67 {
68     Comision_por_Zona_donde_trabaja = Salario_Bruto_1 * 0.15;
69 }
70 else if ("Este".equals(Zona))
71 {
72     Comision_por_Zona_donde_trabaja = Salario_Bruto_1 * 0.20;
73 }
74 else if ("Oeste".equals(Zona))
75 {
76     Comision_por_Zona_donde_trabaja = Salario_Bruto_1 * 0.25;
77 }
78 else if ("NorEste".equals(Zona))
79 {
80     Comision_por_Zona_donde_trabaja = Salario_Bruto_1 * 0.25;
81 }
82 else if ("NorOeste".equals(Zona) || "SurEste".equals(Zona) || "NorOeste".equals(Zona) || "SurOeste".equals(Zona))
83 {
84     Comision_por_Zona_donde_trabaja = Salario_Bruto_1 * 0.25;
85 }
86 else if ("Otra".equals(Zona))
87 {
88     Comision_por_Zona_donde_trabaja = Salario_Bruto_1 * 0.00;
89 }
90 Salario_Bruto_2 = Salario_Bruto_1 + Comision_por_Zona_donde_trabaja;
```

Clase "Planilla".

```
Man_Menu.java x Afiliacion.java x Empleado.java x Producto.java x Promocion.java x Encuesta.java x Facturacion.java x Planilla.java x Inventario.java x Info_Productos.java x Estados.java x
Source History
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
System.out.println(" \n ");
System.out.println("
Planilla: \n");
System.out.println(" \n ");
System.out.println(" \n ");
System.out.println(" Estimado empleado: " +Nombre_Empleado+ " " +_1er_Apellido+ " " +_2do_Apellido+ ". El desglose completo de este mes de su salario es: \n");
System.out.println(" \n ");
System.out.println("
Código: " +Codigo_Empleado+ " \n ");
System.out.println("
Cédula: " +Cedula_Empleado+ " \n ");
System.out.println("
Teléfono: " +Telefono_Empleado+ " \n ");
System.out.println(" \n ");
System.out.println("
Salario por Hora: " +Salario_por_Hora+ " \n ");
System.out.println("
Horas Trabajadas: " +Horas_Trabajadas+ " \n ");
System.out.println("
Horas Extras: " +Horas_Extras+ " \n ");
System.out.println("
Salario Bruto 1: " +Salario_Bruto_1+ " \n ");
System.out.println(" \n ");
System.out.println("
Comisión por Zona donde trabaja: " +Comision_por_Zona_donde_trabaja+ " \n ");
System.out.println("
Salario Bruto 2: " +Salario_Bruto_2+ " \n ");
System.out.println(" \n ");
System.out.println("
Deducción de la Caja (CCSS): " +Deducccion_de_la_Caja_CCSS+ " \n ");
System.out.println("
Deducción del Banco Popular (BP): " +Deducccion_del_Banco_Popular_BPDC+ " \n ");
System.out.println("
Deducción del Sindicato: " +Deducccion_del_Sindicato+ " \n ");
System.out.println("
Deducción del Ahorro Personal: " +Deducccion_del_Ahorro_Personal+ " \n ");
System.out.println("
Deducción del Ahorro Navideño: " +Deducccion_del_Ahorro_Navidenio+ " \n ");
System.out.println("
Total de Deducciones: " +Total_de_Deducciones+ " \n ");
System.out.println(" \n ");
System.out.println("
Salario Neto: " +Salario_Neto+ " \n ");
System.out.println(" \n ");
}
```

```
Main_Menu.java x Afiliacion.java x Empleado.java x Producto.java x Promocion.java x Encuesta.java x Facturacion.java x Planilla.java x Inventario.java x Info_Productos.java x Estados.java x
Source History
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
JOptionPane.showMessageDialog(null, "
Planilla del Empleado: \n "
+ " \n "
+ " Estimado empleado: " +Nombre_Empleado+ " " +_1er_Apellido+ " " +_2do_Apellido+ ". El desglose completo de este mes de su salario es: \n "
+ " \n "
+ "
Código: " +Codigo_Empleado+ " \n "
+ "
Cédula: " +Cedula_Empleado+ " \n "
+ "
Teléfono: " +Telefono_Empleado+ " \n "
+ " \n "
+ "
Salario por Hora: " +Salario_por_Hora
+ "
Horas Trabajadas: " +Horas_Trabajadas
+ "
Horas Extras: " +Horas_Extras
+ "
Salario Bruto 1: " +Salario_Bruto_1+ " \n "
+ " \n "
+ "
Zona donde trabaja: " +Zona
+ "
Comisión por Zona donde trabaja: " +Comision_por_Zona_donde_trabaja
+ "
Salario Bruto 2: " +Salario_Bruto_2+ " \n "
+ " \n "
+ "
Deducción de la Caja (CCSS): " +Deducccion_de_la_Caja_CCSS
+ "
Deducción del Banco Popular (BP): " +Deducccion_del_Banco_Popular_BPDC
+ "
Deducción del Sindicato: " +Deducccion_del_Sindicato
+ "
Deducción del Ahorro Personal: " +Deducccion_del_Ahorro_Personal
+ "
Deducción del Ahorro Navideño: " +Deducccion_del_Ahorro_Navidenio
+ "
Total de Deducciones: " +Total_de_Deducciones+ " \n "
+ " \n "
+ "
Salario Neto: " +Salario_Neto+ " \n "
+ " \n ");
}
```



Clase “Planilla”.

```
91 Deduccion_de_la_Caja_CCSS = Salario_Bruto_2 * 0.10;
92 Deduccion_del_Banco_Popular_BPDC = Salario_Bruto_2 * 0.01;
93 Respuesta_1 = JOptionPane.showInputDialog(" Por favor indique si el Empleado pertenece a algún Sindicato: (Si o No) ");
94 if("Si".equals(Respuesta_1))
95 {
96     Deduccion_del_Sindicato = Salario_Bruto_2 * 0.02;
97 }
98 else
99 {
100     Deduccion_del_Sindicato = Salario_Bruto_2 * 0.00;
101 }
102 Respuesta_2 = JOptionPane.showInputDialog(" Por favor indique si al Empleado le gustaria un Ahorro Personal: (Si o No) ");
103 if("Si".equals(Respuesta_2))
104 {
105     Deduccion_del_Ahorro_Personal = Salario_Bruto_2 * 0.02;
106 }
107 else
108 {
109     Deduccion_del_Ahorro_Personal = Salario_Bruto_2 * 0.00;
110 }
111 Respuesta_3 = JOptionPane.showInputDialog(" Por favor indique si al Empleado le gustaria un Ahorro Navideño: (Si o No) ");
112 if("Si".equals(Respuesta_3))
113 {
114     Deduccion_del_Ahorro_Navideño = Salario_Bruto_2 * 0.02;
115 }
116 else
117 {
118     Deduccion_del_Ahorro_Navideño = Salario_Bruto_2 * 0.00;
119 }
120 Total_de_Deducciones = Deduccion_de_la_Caja_CCSS + Deduccion_del_Banco_Popular_BPDC + Deduccion_del_Sindicato + Deduccion_del_Ahorro_Personal + Deduccion_del_Ahorro_Navideño;
121 Salario_Neto = Salario_Bruto_2 - Total_de_Deducciones;
```

```
Planilla:

Estimado empleado: Esteban Ugalde Rojas. El desglose completo de este mes de su salario es:

Código: 179JeanpiEUR101112
Cédula: 118360459
Teléfono: 84199675

Salario por Hora: 5000.0
Horas Trabajadas: 200
Horas Extras: 8.0
Salario Bruto 1: 1020000.0

Zona donde trabaja: NorOeste
Comisión por Zona donde trabaja: 357000.0
Salario Bruto 2: 1377000.0

Zona donde trabaja: NorOeste
Comisión por Zona donde trabaja: 357000.0
Salario Bruto 2: 1377000.0

Deducción de la Caja (CCSS): 137700.0
Deducción del Banco Popular (BP): 13770.0
Deducción del Sindicato: 0.0
Deducción del Ahorro Personal: 0.0
Deducción del Ahorro Navideño: 0.0
Total de Deducciones: 151470.0

Salario Neto: 1225530.0
```


Clase “Inventario”.

En esta clase al selecciona la opción 7 se prosigue a digitar e imprimir los datos y la disponibilidad de los productos en el inventario.

Esta clase después de digitar los datos que solicita esta clase, captura los datos almacenados en la clase “Info_Productos”.

Además esta clase puede capturar el estado desde las clases “Info_Productos” y “Estados”, si yo digite en los datos que la cantidad disponible del producto es 0 me imprime todos los que ingrese y me imprime que el producto esta agotado, de lo contrario me imprime que el producto esta disponible.

```
Start Page x Main_Menu.java x Afiliacion.java x Empleado.java x Producto.java x Promocion.java x Encuesta.java x Facturacion.java x Planilla.java x Inventario.java x Info_Productos.java x Estados.java x
Source History

1 package superfide;
2
3 import javax.swing.JOptionPane;
4
5 public class Inventario
6 {
7     int X = 0;
8     int Y = 0;
9     int FILA = 0;
10    int COLUMNA = 0;
11    Info_Productos CantidadRegistrada[][] = new Info_Productos[1][1];
12
13    public void Inventario()
14    {
15        // En este METODO se llenara el arreglo BIDIMENSIONAL (MATRIZ) y tambien se le dará tamaño al arreglo BIDIMENSIONAL (MATRIZ) ARREGLO DE DOS DIMENSIONES (FILAS Y COLUMNAS).
16        // LAS FILAS SON VERTICALES Y A NIVEL DE COORDENADAS SE REPRESENTAN EN EL EJE X DESDE X POSITIVO A X NEGATIVO.
17        // LAS COLUMNAS SON HORIZONTALES Y A NIVEL DE COORDENADAS SE REPRESENTAN EN EL EJE Y DESDE Y POSITIVO A Y NEGATIVO.
18
19        for(int X = 0; X < 1; X++)
20        {
21            for(int Y = 0; Y < 1; Y++)
22            {
23                // Aqui se llenan uno a uno el arreglo BIDIMENSIONAL (MATRIZ) ARREGLO DE DOS DIMENSIONES.
24
25                Info_Productos InfoProd = new Info_Productos();
26                InfoProd.setCodigo(JOptionPane.showInputDialog(" Por favor digite el Código: "));
27                InfoProd.setNombre(JOptionPane.showInputDialog(" Por favor digite el Nombre: "));
28                InfoProd.setFecha_de_Produccion_o_Fabricacion(JOptionPane.showInputDialog(" Por favor digite la Fecha de Producción o de Fabricación: "));
29                InfoProd.setFecha_de_Vencimiento(JOptionPane.showInputDialog(" Por favor digite la Fecha de Vencimiento: "));
30                InfoProd.setPrecio(Double.parseDouble(JOptionPane.showInputDialog(" Por favor digite el Precio de una unidad: ")));
31                InfoProd.setCantidad(Long.parseLong(JOptionPane.showInputDialog(" Por favor digite la Cantidad de Unidades Disponibles: ")));
32                InfoProd.setPasillo(Long.parseLong(JOptionPane.showInputDialog(" Por favor digite el Pasillo donde se encuentra el producto: ")));
33
34                CantidadRegistrada[X][Y] = InfoProd;
35            }
36        }
37    }
38
39    for(int X = 0; X < 1; X++)
40    {
41        for(int Y = 0; Y < 1; Y++)
42        {
43            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
44            {
45                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
46                JOptionPane.showMessageDialog(null, " \n "
47                    + "
48                    + "
49                    + "
50                    + "
51                    + "
52                    + "
53                    + "
54                    + "
55                    + " \n ");
56
57                System.out.println(" \n ");
58                System.out.println("
59                System.out.println("
60                System.out.println("
61                System.out.println("
62                System.out.println("
63                System.out.println("
64                System.out.println("
65                System.out.println("
66                System.out.println("
67                System.out.println(" \n ");
68            }
69        }
70    }
71
72    for(int X = 0; X < 1; X++)
73    {
74        for(int Y = 0; Y < 1; Y++)
75        {
76            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
77            {
78                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
79                JOptionPane.showMessageDialog(null, " \n "
80                    + "
81                    + "
82                    + "
83                    + "
84                    + "
85                    + "
86                    + "
87                    + "
88                    + " \n ");
89
90                System.out.println(" \n ");
91                System.out.println("
92                System.out.println("
93                System.out.println("
94                System.out.println("
95                System.out.println("
96                System.out.println("
97                System.out.println("
98                System.out.println("
99                System.out.println(" \n ");
100            }
101        }
102    }
103
104    for(int X = 0; X < 1; X++)
105    {
106        for(int Y = 0; Y < 1; Y++)
107        {
108            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
109            {
110                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
111                JOptionPane.showMessageDialog(null, " \n "
112                    + "
113                    + "
114                    + "
115                    + "
116                    + "
117                    + "
118                    + "
119                    + "
120                    + " \n ");
121
122                System.out.println(" \n ");
123                System.out.println("
124                System.out.println("
125                System.out.println("
126                System.out.println("
127                System.out.println("
128                System.out.println("
129                System.out.println("
130                System.out.println("
131                System.out.println(" \n ");
132            }
133        }
134    }
135
136    for(int X = 0; X < 1; X++)
137    {
138        for(int Y = 0; Y < 1; Y++)
139        {
140            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
141            {
142                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
143                JOptionPane.showMessageDialog(null, " \n "
144                    + "
145                    + "
146                    + "
147                    + "
148                    + "
149                    + "
150                    + "
151                    + "
152                    + " \n ");
153
154                System.out.println(" \n ");
155                System.out.println("
156                System.out.println("
157                System.out.println("
158                System.out.println("
159                System.out.println("
160                System.out.println("
161                System.out.println("
162                System.out.println("
163                System.out.println(" \n ");
164            }
165        }
166    }
167
168    for(int X = 0; X < 1; X++)
169    {
170        for(int Y = 0; Y < 1; Y++)
171        {
172            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
173            {
174                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
175                JOptionPane.showMessageDialog(null, " \n "
176                    + "
177                    + "
178                    + "
179                    + "
180                    + "
181                    + "
182                    + "
183                    + "
184                    + " \n ");
185
186                System.out.println(" \n ");
187                System.out.println("
188                System.out.println("
189                System.out.println("
190                System.out.println("
191                System.out.println("
192                System.out.println("
193                System.out.println("
194                System.out.println("
195                System.out.println(" \n ");
196            }
197        }
198    }
199
200    for(int X = 0; X < 1; X++)
201    {
202        for(int Y = 0; Y < 1; Y++)
203        {
204            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
205            {
206                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
207                JOptionPane.showMessageDialog(null, " \n "
208                    + "
209                    + "
210                    + "
211                    + "
212                    + "
213                    + "
214                    + "
215                    + "
216                    + " \n ");
217
218                System.out.println(" \n ");
219                System.out.println("
220                System.out.println("
221                System.out.println("
222                System.out.println("
223                System.out.println("
224                System.out.println("
225                System.out.println("
226                System.out.println("
227                System.out.println(" \n ");
228            }
229        }
230    }
231
232    for(int X = 0; X < 1; X++)
233    {
234        for(int Y = 0; Y < 1; Y++)
235        {
236            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
237            {
238                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
239                JOptionPane.showMessageDialog(null, " \n "
240                    + "
241                    + "
242                    + "
243                    + "
244                    + "
245                    + "
246                    + "
247                    + "
248                    + " \n ");
249
250                System.out.println(" \n ");
251                System.out.println("
252                System.out.println("
253                System.out.println("
254                System.out.println("
255                System.out.println("
256                System.out.println("
257                System.out.println("
258                System.out.println("
259                System.out.println(" \n ");
260            }
261        }
262    }
263
264    for(int X = 0; X < 1; X++)
265    {
266        for(int Y = 0; Y < 1; Y++)
267        {
268            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
269            {
270                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
271                JOptionPane.showMessageDialog(null, " \n "
272                    + "
273                    + "
274                    + "
275                    + "
276                    + "
277                    + "
278                    + "
279                    + "
280                    + " \n ");
281
282                System.out.println(" \n ");
283                System.out.println("
284                System.out.println("
285                System.out.println("
286                System.out.println("
287                System.out.println("
288                System.out.println("
289                System.out.println("
290                System.out.println("
291                System.out.println(" \n ");
292            }
293        }
294    }
295
296    for(int X = 0; X < 1; X++)
297    {
298        for(int Y = 0; Y < 1; Y++)
299        {
300            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
301            {
302                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
303                JOptionPane.showMessageDialog(null, " \n "
304                    + "
305                    + "
306                    + "
307                    + "
308                    + "
309                    + "
310                    + "
311                    + "
312                    + " \n ");
313
314                System.out.println(" \n ");
315                System.out.println("
316                System.out.println("
317                System.out.println("
318                System.out.println("
319                System.out.println("
320                System.out.println("
321                System.out.println("
322                System.out.println("
323                System.out.println(" \n ");
324            }
325        }
326    }
327
328    for(int X = 0; X < 1; X++)
329    {
330        for(int Y = 0; Y < 1; Y++)
331        {
332            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
333            {
334                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
335                JOptionPane.showMessageDialog(null, " \n "
336                    + "
337                    + "
338                    + "
339                    + "
340                    + "
341                    + "
342                    + "
343                    + "
344                    + " \n ");
345
346                System.out.println(" \n ");
347                System.out.println("
348                System.out.println("
349                System.out.println("
350                System.out.println("
351                System.out.println("
352                System.out.println("
353                System.out.println("
354                System.out.println("
355                System.out.println(" \n ");
356            }
357        }
358    }
359
360    for(int X = 0; X < 1; X++)
361    {
362        for(int Y = 0; Y < 1; Y++)
363        {
364            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
365            {
366                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
367                JOptionPane.showMessageDialog(null, " \n "
368                    + "
369                    + "
370                    + "
371                    + "
372                    + "
373                    + "
374                    + "
375                    + "
376                    + " \n ");
377
378                System.out.println(" \n ");
379                System.out.println("
380                System.out.println("
381                System.out.println("
382                System.out.println("
383                System.out.println("
384                System.out.println("
385                System.out.println("
386                System.out.println("
387                System.out.println(" \n ");
388            }
389        }
390    }
391
392    for(int X = 0; X < 1; X++)
393    {
394        for(int Y = 0; Y < 1; Y++)
395        {
396            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
397            {
398                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
399                JOptionPane.showMessageDialog(null, " \n "
400                    + "
401                    + "
402                    + "
403                    + "
404                    + "
405                    + "
406                    + "
407                    + "
408                    + " \n ");
409
410                System.out.println(" \n ");
411                System.out.println("
412                System.out.println("
413                System.out.println("
414                System.out.println("
415                System.out.println("
416                System.out.println("
417                System.out.println("
418                System.out.println("
419                System.out.println(" \n ");
420            }
421        }
422    }
423
424    for(int X = 0; X < 1; X++)
425    {
426        for(int Y = 0; Y < 1; Y++)
427        {
428            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
429            {
430                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
431                JOptionPane.showMessageDialog(null, " \n "
432                    + "
433                    + "
434                    + "
435                    + "
436                    + "
437                    + "
438                    + "
439                    + "
440                    + " \n ");
441
442                System.out.println(" \n ");
443                System.out.println("
444                System.out.println("
445                System.out.println("
446                System.out.println("
447                System.out.println("
448                System.out.println("
449                System.out.println("
450                System.out.println("
451                System.out.println(" \n ");
452            }
453        }
454    }
455
456    for(int X = 0; X < 1; X++)
457    {
458        for(int Y = 0; Y < 1; Y++)
459        {
460            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
461            {
462                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
463                JOptionPane.showMessageDialog(null, " \n "
464                    + "
465                    + "
466                    + "
467                    + "
468                    + "
469                    + "
470                    + "
471                    + "
472                    + " \n ");
473
474                System.out.println(" \n ");
475                System.out.println("
476                System.out.println("
477                System.out.println("
478                System.out.println("
479                System.out.println("
480                System.out.println("
481                System.out.println("
482                System.out.println("
483                System.out.println(" \n ");
484            }
485        }
486    }
487
488    for(int X = 0; X < 1; X++)
489    {
490        for(int Y = 0; Y < 1; Y++)
491        {
492            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
493            {
494                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
495                JOptionPane.showMessageDialog(null, " \n "
496                    + "
497                    + "
498                    + "
499                    + "
500                    + "
501                    + "
502                    + "
503                    + "
504                    + " \n ");
505
506                System.out.println(" \n ");
507                System.out.println("
508                System.out.println("
509                System.out.println("
510                System.out.println("
511                System.out.println("
512                System.out.println("
513                System.out.println("
514                System.out.println("
515                System.out.println(" \n ");
516            }
517        }
518    }
519
520    for(int X = 0; X < 1; X++)
521    {
522        for(int Y = 0; Y < 1; Y++)
523        {
524            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
525            {
526                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
527                JOptionPane.showMessageDialog(null, " \n "
528                    + "
529                    + "
530                    + "
531                    + "
532                    + "
533                    + "
534                    + "
535                    + "
536                    + " \n ");
537
538                System.out.println(" \n ");
539                System.out.println("
540                System.out.println("
541                System.out.println("
542                System.out.println("
543                System.out.println("
544                System.out.println("
545                System.out.println("
546                System.out.println("
547                System.out.println(" \n ");
548            }
549        }
550    }
551
552    for(int X = 0; X < 1; X++)
553    {
554        for(int Y = 0; Y < 1; Y++)
555        {
556            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
557            {
558                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
559                JOptionPane.showMessageDialog(null, " \n "
560                    + "
561                    + "
562                    + "
563                    + "
564                    + "
565                    + "
566                    + "
567                    + "
568                    + " \n ");
569
570                System.out.println(" \n ");
571                System.out.println("
572                System.out.println("
573                System.out.println("
574                System.out.println("
575                System.out.println("
576                System.out.println("
577                System.out.println("
578                System.out.println("
579                System.out.println(" \n ");
580            }
581        }
582    }
583
584    for(int X = 0; X < 1; X++)
585    {
586        for(int Y = 0; Y < 1; Y++)
587        {
588            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
589            {
590                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
591                JOptionPane.showMessageDialog(null, " \n "
592                    + "
593                    + "
594                    + "
595                    + "
596                    + "
597                    + "
598                    + "
599                    + "
600                    + " \n ");
601
602                System.out.println(" \n ");
603                System.out.println("
604                System.out.println("
605                System.out.println("
606                System.out.println("
607                System.out.println("
608                System.out.println("
609                System.out.println("
610                System.out.println("
611                System.out.println(" \n ");
612            }
613        }
614    }
615
616    for(int X = 0; X < 1; X++)
617    {
618        for(int Y = 0; Y < 1; Y++)
619        {
620            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
621            {
622                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
623                JOptionPane.showMessageDialog(null, " \n "
624                    + "
625                    + "
626                    + "
627                    + "
628                    + "
629                    + "
630                    + "
631                    + "
632                    + " \n ");
633
634                System.out.println(" \n ");
635                System.out.println("
636                System.out.println("
637                System.out.println("
638                System.out.println("
639                System.out.println("
640                System.out.println("
641                System.out.println("
642                System.out.println("
643                System.out.println(" \n ");
644            }
645        }
646    }
647
648    for(int X = 0; X < 1; X++)
649    {
650        for(int Y = 0; Y < 1; Y++)
651        {
652            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
653            {
654                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
655                JOptionPane.showMessageDialog(null, " \n "
656                    + "
657                    + "
658                    + "
659                    + "
660                    + "
661                    + "
662                    + "
663                    + "
664                    + " \n ");
665
666                System.out.println(" \n ");
667                System.out.println("
668                System.out.println("
669                System.out.println("
670                System.out.println("
671                System.out.println("
672                System.out.println("
673                System.out.println("
674                System.out.println("
675                System.out.println(" \n ");
676            }
677        }
678    }
679
680    for(int X = 0; X < 1; X++)
681    {
682        for(int Y = 0; Y < 1; Y++)
683        {
684            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
685            {
686                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
687                JOptionPane.showMessageDialog(null, " \n "
688                    + "
689                    + "
690                    + "
691                    + "
692                    + "
693                    + "
694                    + "
695                    + "
696                    + " \n ");
697
698                System.out.println(" \n ");
699                System.out.println("
700                System.out.println("
701                System.out.println("
702                System.out.println("
703                System.out.println("
704                System.out.println("
705                System.out.println("
706                System.out.println("
707                System.out.println(" \n ");
708            }
709        }
710    }
711
712    for(int X = 0; X < 1; X++)
713    {
714        for(int Y = 0; Y < 1; Y++)
715        {
716            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
717            {
718                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
719                JOptionPane.showMessageDialog(null, " \n "
720                    + "
721                    + "
722                    + "
723                    + "
724                    + "
725                    + "
726                    + "
727                    + "
728                    + " \n ");
729
730                System.out.println(" \n ");
731                System.out.println("
732                System.out.println("
733                System.out.println("
734                System.out.println("
735                System.out.println("
736                System.out.println("
737                System.out.println("
738                System.out.println("
739                System.out.println(" \n ");
740            }
741        }
742    }
743
744    for(int X = 0; X < 1; X++)
745    {
746        for(int Y = 0; Y < 1; Y++)
747        {
748            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
749            {
750                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
751                JOptionPane.showMessageDialog(null, " \n "
752                    + "
753                    + "
754                    + "
755                    + "
756                    + "
757                    + "
758                    + "
759                    + "
760                    + " \n ");
761
762                System.out.println(" \n ");
763                System.out.println("
764                System.out.println("
765                System.out.println("
766                System.out.println("
767                System.out.println("
768                System.out.println("
769                System.out.println("
770                System.out.println("
771                System.out.println(" \n ");
772            }
773        }
774    }
775
776    for(int X = 0; X < 1; X++)
777    {
778        for(int Y = 0; Y < 1; Y++)
779        {
780            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
781            {
782                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
783                JOptionPane.showMessageDialog(null, " \n "
784                    + "
785                    + "
786                    + "
787                    + "
788                    + "
789                    + "
790                    + "
791                    + "
792                    + " \n ");
793
794                System.out.println(" \n ");
795                System.out.println("
796                System.out.println("
797                System.out.println("
798                System.out.println("
799                System.out.println("
800                System.out.println("
801                System.out.println("
802                System.out.println("
803                System.out.println(" \n ");
804            }
805        }
806    }
807
808    for(int X = 0; X < 1; X++)
809    {
810        for(int Y = 0; Y < 1; Y++)
811        {
812            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
813            {
814                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
815                JOptionPane.showMessageDialog(null, " \n "
816                    + "
817                    + "
818                    + "
819                    + "
820                    + "
821                    + "
822                    + "
823                    + "
824                    + " \n ");
825
826                System.out.println(" \n ");
827                System.out.println("
828                System.out.println("
829                System.out.println("
830                System.out.println("
831                System.out.println("
832                System.out.println("
833                System.out.println("
834                System.out.println("
835                System.out.println(" \n ");
836            }
837        }
838    }
839
840    for(int X = 0; X < 1; X++)
841    {
842        for(int Y = 0; Y < 1; Y++)
843        {
844            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
845            {
846                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
847                JOptionPane.showMessageDialog(null, " \n "
848                    + "
849                    + "
850                    + "
851                    + "
852                    + "
853                    + "
854                    + "
855                    + "
856                    + " \n ");
857
858                System.out.println(" \n ");
859                System.out.println("
860                System.out.println("
861                System.out.println("
862                System.out.println("
863                System.out.println("
864                System.out.println("
865                System.out.println("
866                System.out.println("
867                System.out.println(" \n ");
868            }
869        }
870    }
871
872    for(int X = 0; X < 1; X++)
873    {
874        for(int Y = 0; Y < 1; Y++)
875        {
876            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
877            {
878                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
879                JOptionPane.showMessageDialog(null, " \n "
880                    + "
881                    + "
882                    + "
883                    + "
884                    + "
885                    + "
886                    + "
887                    + "
888                    + " \n ");
889
890                System.out.println(" \n ");
891                System.out.println("
892                System.out.println("
893                System.out.println("
894                System.out.println("
895                System.out.println("
896                System.out.println("
897                System.out.println("
898                System.out.println("
899                System.out.println(" \n ");
900            }
901        }
902    }
903
904    for(int X = 0; X < 1; X++)
905    {
906        for(int Y = 0; Y < 1; Y++)
907        {
908            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
909            {
910                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
911                JOptionPane.showMessageDialog(null, " \n "
912                    + "
913                    + "
914                    + "
915                    + "
916                    + "
917                    + "
918                    + "
919                    + "
920                    + " \n ");
921
922                System.out.println(" \n ");
923                System.out.println("
924                System.out.println("
925                System.out.println("
926                System.out.println("
927                System.out.println("
928                System.out.println("
929                System.out.println("
930                System.out.println("
931                System.out.println(" \n ");
932            }
933        }
934    }
935
936    for(int X = 0; X < 1; X++)
937    {
938        for(int Y = 0; Y < 1; Y++)
939        {
940            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
941            {
942                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
943                JOptionPane.showMessageDialog(null, " \n "
944                    + "
945                    + "
946                    + "
947                    + "
948                    + "
949                    + "
950                    + "
951                    + "
952                    + " \n ");
953
954                System.out.println(" \n ");
955                System.out.println("
956                System.out.println("
957                System.out.println("
958                System.out.println("
959                System.out.println("
960                System.out.println("
961                System.out.println("
962                System.out.println("
963                System.out.println(" \n ");
964            }
965        }
966    }
967
968    for(int X = 0; X < 1; X++)
969    {
970        for(int Y = 0; Y < 1; Y++)
971        {
972            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
973            {
974                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
975                JOptionPane.showMessageDialog(null, " \n "
976                    + "
977                    + "
978                    + "
979                    + "
980                    + "
981                    + "
982                    + "
983                    + "
984                    + " \n ");
985
986                System.out.println(" \n ");
987                System.out.println("
988                System.out.println("
989                System.out.println("
990                System.out.println("
991                System.out.println("
992                System.out.println("
993                System.out.println("
994                System.out.println("
995                System.out.println(" \n ");
996            }
997        }
998    }
999
1000    for(int X = 0; X < 1; X++)
1001    {
1002        for(int Y = 0; Y < 1; Y++)
1003        {
1004            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
1005            {
1006                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
1007                JOptionPane.showMessageDialog(null, " \n "
1008                    + "
1009                    + "
1010                    + "
1011                    + "
1012                    + "
1013                    + "
1014                    + "
1015                    + "
1016                    + " \n ");
1017
1018                System.out.println(" \n ");
1019                System.out.println("
1020                System.out.println("
1021                System.out.println("
1022                System.out.println("
1023                System.out.println("
1024                System.out.println("
1025                System.out.println("
1026                System.out.println("
1027                System.out.println(" \n ");
1028            }
1029        }
1030    }
1031
1032    for(int X = 0; X < 1; X++)
1033    {
1034        for(int Y = 0; Y < 1; Y++)
1035        {
1036            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
1037            {
1038                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
1039                JOptionPane.showMessageDialog(null, " \n "
1040                    + "
1041                    + "
1042                    + "
1043                    + "
1044                    + "
1045                    + "
1046                    + "
1047                    + "
1048                    + " \n ");
1049
1050                System.out.println(" \n ");
1051                System.out.println("
1052                System.out.println("
1053                System.out.println("
1054                System.out.println("
1055                System.out.println("
1056                System.out.println("
1057                System.out.println("
1058                System.out.println("
1059                System.out.println(" \n ");
1060            }
1061        }
1062    }
1063
1064    for(int X = 0; X < 1; X++)
1065    {
1066        for(int Y = 0; Y < 1; Y++)
1067        {
1068            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
1069            {
1070                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
1071                JOptionPane.showMessageDialog(null, " \n "
1072                    + "
1073                    + "
1074                    + "
1075                    + "
1076                    + "
1077                    + "
1078                    + "
1079                    + "
1080                    + " \n ");
1081
1082                System.out.println(" \n ");
1083                System.out.println("
1084                System.out.println("
1085                System.out.println("
1086                System.out.println("
1087                System.out.println("
1088                System.out.println("
1089                System.out.println("
1090                System.out.println("
1091                System.out.println(" \n ");
1092            }
1093        }
1094    }
1095
1096    for(int X = 0; X < 1; X++)
1097    {
1098        for(int Y = 0; Y < 1; Y++)
1099        {
1100            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
1101            {
1102                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
1103                JOptionPane.showMessageDialog(null, " \n "
1104                    + "
1105                    + "
1106                    + "
1107                    + "
1108                    + "
1109                    + "
1110                    + "
1111                    + "
1112                    + " \n ");
1113
1114                System.out.println(" \n ");
1115                System.out.println("
1116                System.out.println("
1117                System.out.println("
1118                System.out.println("
1119                System.out.println("
1120                System.out.println("
1121                System.out.println("
1122                System.out.println("
1123                System.out.println(" \n ");
1124            }
1125        }
1126    }
1127
1128    for(int X = 0; X < 1; X++)
1129    {
1130        for(int Y = 0; Y < 1; Y++)
1131        {
1132            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
1133            {
1134                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
1135                JOptionPane.showMessageDialog(null, " \n "
1136                    + "
1137                    + "
1138                    + "
1139                    + "
1140                    + "
1141                    + "
1142                    + "
1143                    + "
1144                    + " \n ");
1145
1146                System.out.println(" \n ");
1147                System.out.println("
1148                System.out.println("
1149                System.out.println("
1150                System.out.println("
1151                System.out.println("
1152                System.out.println("
1153                System.out.println("
1154                System.out.println("
1155                System.out.println(" \n ");
1156            }
1157        }
1158    }
1159
1160    for(int X = 0; X < 1; X++)
1161    {
1162        for(int Y = 0; Y < 1; Y++)
1163        {
1164            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
1165            {
1166                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
1167                JOptionPane.showMessageDialog(null, " \n "
1168                    + "
1169                    + "
1170                    + "
1171                    + "
1172                    + "
1173                    + "
1174                    + "
1175                    + "
1176                    + " \n ");
1177
1178                System.out.println(" \n ");
1179                System.out.println("
1180                System.out.println("
1181                System.out.println("
1182                System.out.println("
1183                System.out.println("
1184                System.out.println("
1185                System.out.println("
1186                System.out.println("
1187                System.out.println(" \n ");
1188            }
1189        }
1190    }
1191
1192    for(int X = 0; X < 1; X++)
1193    {
1194        for(int Y = 0; Y < 1; Y++)
1195        {
1196            if(CantidadRegistrada[FILA][COLUMNA].getCantidad() == 0)
1197            {
1198                CantidadRegistrada[FILA][COLUMNA].setEstado(Estados.Agotado);
1199                JOptionPane.showMessageDialog(null, " \n "
1200                    + "
1201                    + "
1202                    + "
1203                    + "
1204                    + "
1205                    + "
1206                    + "
1207                    + "
120
```

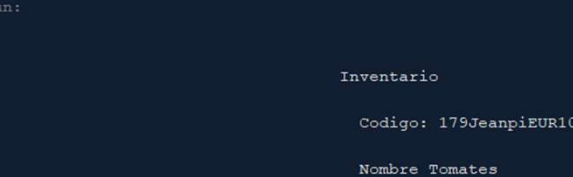
Clase “Inventario”.



```

68     }
69     else
70     {
71         CantidadRegistrada = [FILA][COLUMNA].getEstado(Estados.Disponible);
72         JOptionPane.showMessageDialog(null, " \n "
73             + "
74             Inventario: " + " \n "
75             + "
76             + "CantidadRegistrada [FILA][COLUMNA].getCodigo() + " \n "
77             + "
78             Nombre: " + "CantidadRegistrada [FILA][COLUMNA].getNombre() + " \n "
79             + "
80             Fecha de Producción o Fabricación: " + "CantidadRegistrada [FILA][COLUMNA].getFecha_de_Produccion_o_Fabricacion() + " \n "
81             + "
82             Fecha de Vencimiento: " + "CantidadRegistrada [FILA][COLUMNA].getFecha_de_Vencimiento() + " \n "
83             + "
84             Precio: " + "CantidadRegistrada [FILA][COLUMNA].getPrecio() + " \n "
85             + "
86             Cantidades Disponibles: " + "CantidadRegistrada [FILA][COLUMNA].getCantidad() + " \n "
87             + "
88             Pasillo: " + "CantidadRegistrada [FILA][COLUMNA].getPasillo() + " \n "
89             + "
90             Estado: " + "Estados.Disponible + " \n "
91             + " \n ");
92     }
93
94     System.out.println(" \n ");
95     System.out.println("
96     Inventario: " + " \n ");
97     System.out.println("
98     + "CantidadRegistrada [FILA][COLUMNA].getCodigo() + " \n ");
99     System.out.println("
100    + "CantidadRegistrada [FILA][COLUMNA].getNombre() + " \n ");
101     System.out.println("
102    Fecha de Producción o Fabricación: " + "CantidadRegistrada [FILA][COLUMNA].getFecha_de_Produccion_o_Fabricacion() + " \n ");
103     System.out.println("
104    Fecha de Vencimiento: " + "CantidadRegistrada [FILA][COLUMNA].getFecha_de_Vencimiento() + " \n ");
105     System.out.println("
106    Precio: " + "CantidadRegistrada [FILA][COLUMNA].getPrecio() + " \n ");
107     System.out.println("
108    Cantidades Disponibles: " + "CantidadRegistrada [FILA][COLUMNA].getCantidad() + " \n ");
109     System.out.println("
110    Pasillo: " + "CantidadRegistrada [FILA][COLUMNA].getPasillo() + " \n ");
111     System.out.println("
112    Estado: " + "Estados.Disponible + " \n ");
113     System.out.println(" \n ");
114 }
115 }

```



```
Output - SuperFide (run) x
run:

Inventario

Codigo: 179JeanpiEUR101112

Nombre Tomates

Fecha de Producción o Fabricación: 15/03/2021

Fecha de Vencimiento: 15/04/2021

Precio: 500.0

Cantidades Disponibles: 999000

Pasillo: 10

Estado: Disponible
```

Clase Info_Productos”.

```
1 package superfide;
2
3
4 public class Info_Productos
5 {
6     private StringCodigo;
7     private String Nombre;
8     private String Fecha_de_Produccion_o_Fabricacion;
9     private String Fecha_de_Vencimiento;
10    private long Cantidad;
11    private double Precio;
12    private long Pasillo;
13    private Estados Estado;
14
15    public Info_Productos()
16    {
17        Codigo = "";
18        Nombre = "";
19        Fecha_de_Produccion_o_Fabricacion = "";
20        Fecha_de_Vencimiento = "";
21        Cantidad = 0;
22        Precio = 0;
23        Pasillo = 0;
24        Estado = Estados.Disponible;
25    }
26
27    public String getCodigo()
28    {
29        return Codigo;
30    }
31
32    public void setCodigo(String Codigo)
33    {
34        this.Codigo = Codigo;
35    }
36
```

```
85    }
86
87    public long getPasillo()
88    {
89        return Pasillo;
90    }
91
92    public void setPasillo(long Pasillo)
93    {
94        this.Pasillo = Pasillo;
95    }
96
97    public Estados getEstado()
```

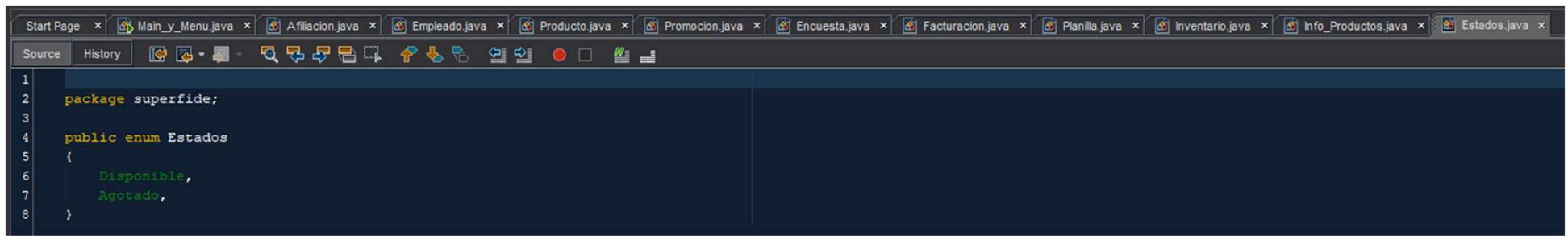
Desde esta clase se pueden capturar los datos y la disponibilidad de los productos, digitados y capturados en la clase “Inventario”.

Además la clase Inventario puede capturar el estado desde esta clase y la clase “Estados”, si yo digite en los datos que la cantidad disponible del producto es 0 me imprime todos los que ingrese y me imprime que el producto esta agotado, de lo contrario me imprime que el producto esta disponible.

Clase “Info_Productos”.

```
Start Page x Main_Menu.java x Afiliacion.java x Empleado.java x Producto.java x Promocion.java x Encuesta.java x Facturacion.java x Planilla.java x Inventario.java x Info_Productos.java x
Source History
37 public String getNombre()
38 {
39     return Nombre;
40 }
41
42 public void setNombre(String Nombre)
43 {
44     this.Nombre = Nombre;
45 }
46
47 public String getFecha_de_Produccion_o_Fabricacion()
48 {
49     return Fecha_de_Produccion_o_Fabricacion;
50 }
51
52 public void setFecha_de_Produccion_o_Fabricacion(String Fecha_de_Produccion_o_Fabricacion)
53 {
54     this.Fecha_de_Produccion_o_Fabricacion = Fecha_de_Produccion_o_Fabricacion;
55 }
56
57 public String getFecha_de_Vencimiento()
58 {
59     return Fecha_de_Vencimiento;
60 }
61
62 public void setFecha_de_Vencimiento(String Fecha_de_Vencimiento)
63 {
64     this.Fecha_de_Vencimiento = Fecha_de_Vencimiento;
65 }
66
67 public void setCantidad(long Cantidad)
68 {
69     this.Cantidad = Cantidad;
70 }
71
72 public long getCantidad()
73 {
74     return Cantidad;
75 }
76
```

Clase “Estados”.



```
1 package superfide;
2
3
4 public enum Estados
5 {
6     Disponible,
7     Agotado,
8 }
```

Desde esta clase se puede capturar la disponibilidad de los productos, digitados y capturados en la clase “Inventario”. Además la clase Inventario puede capturar el estado desde la clase “Info_Productos” y esta clase, si yo digite en los datos que la cantidad disponible del producto es 0 me imprime todos los que ingrese y me imprime que el producto está agotado, de lo contrario me imprime que el producto está disponible.




Alcances.

Logramos que programa fuera accesible para todos los empleados de la pulpería.

Logramos que cliente pudiera comprar cualquier cantidad de productos.

Logramos calcular de una manera más simple los precios, las planillas de salarios y deducciones.






Limitaciones.

En clase inventario y la clase Planilla la hicimos para un solo empleado, ya que implementar que ambas clases fueran accesibles para múltiples empleados requería otros temas mas complicados y no visto en este curso.

En el inventario podíamos haber capturado la información de cualquier cantidad de productos gracias al arreglo unidimensional (vector) de objetos, pero usamos arreglos bidimensionales (Matrices) de objetos ya que el proyecto requería matrices para su evaluación correcta y casi todo el proyecto tenía solamente vectores en casi todas las clases. Los requisitos eran tener una matriz y un arreglo de objetos pero los mezclamos para que fuera mas fácil la programación y además intentamos averiguar como capturar en el inventario cualquier cantidad de productos mediante matrices de objetos pero era mas difícil la programación ya que se requieren leer filas y columnas y los vectores solo pueden leer una fila o una columna.

También antes de implementar matrices de objetos en clase Inventario intentamos buscar videos pero ahí usaban arreglos dinámicos y/o ArrayLists y este no lo hemos visto porque estos temas se ven en otro curso.

Tampoco pudimos que el inventario se actualizara dependiendo de las compra de los clientes, lo tuvimos que actualizar manualmente digitando las cantidades disponibles quitando las unidades compradas por cliente ya que hacer por automático requería programación mas compleja y requería usar archivos (files) y no lo vimos en este curso porque se ve en otro curso





Conclusiones.

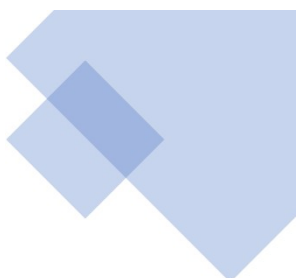

- Logramos basar nuestro proyecto en una pulpería o un minisúper, ya que es algo muy común en nuestras comunidades y se nos hacia muy interesante crear un programa para estas.
 - Logramos enfocar este proyecto para los trabajadores, para así crear un ambiente de trabajo más fácil para ellos a la hora de realizar varias funciones que los clientes necesitan, ya sea realizar un pago, validar un cupón o un descuento etc.
 - Logramos decidir nombrar nuestro proyecto como “Mini Super Fidélitas”.
- 
- 



Diagrama de clases (Primer Avance)

