

# Programación Orientada a Eventos

## Taller #1

Edwar Yamir Forero Blanco

Código:202259664

Juan Eduardo Calderón Jaramillo

Código:202259671

Tercer Semestre

Tecnología en desarrollo de software-2724

Universidad del valle

Tuluá-Valle

Agosto 2022

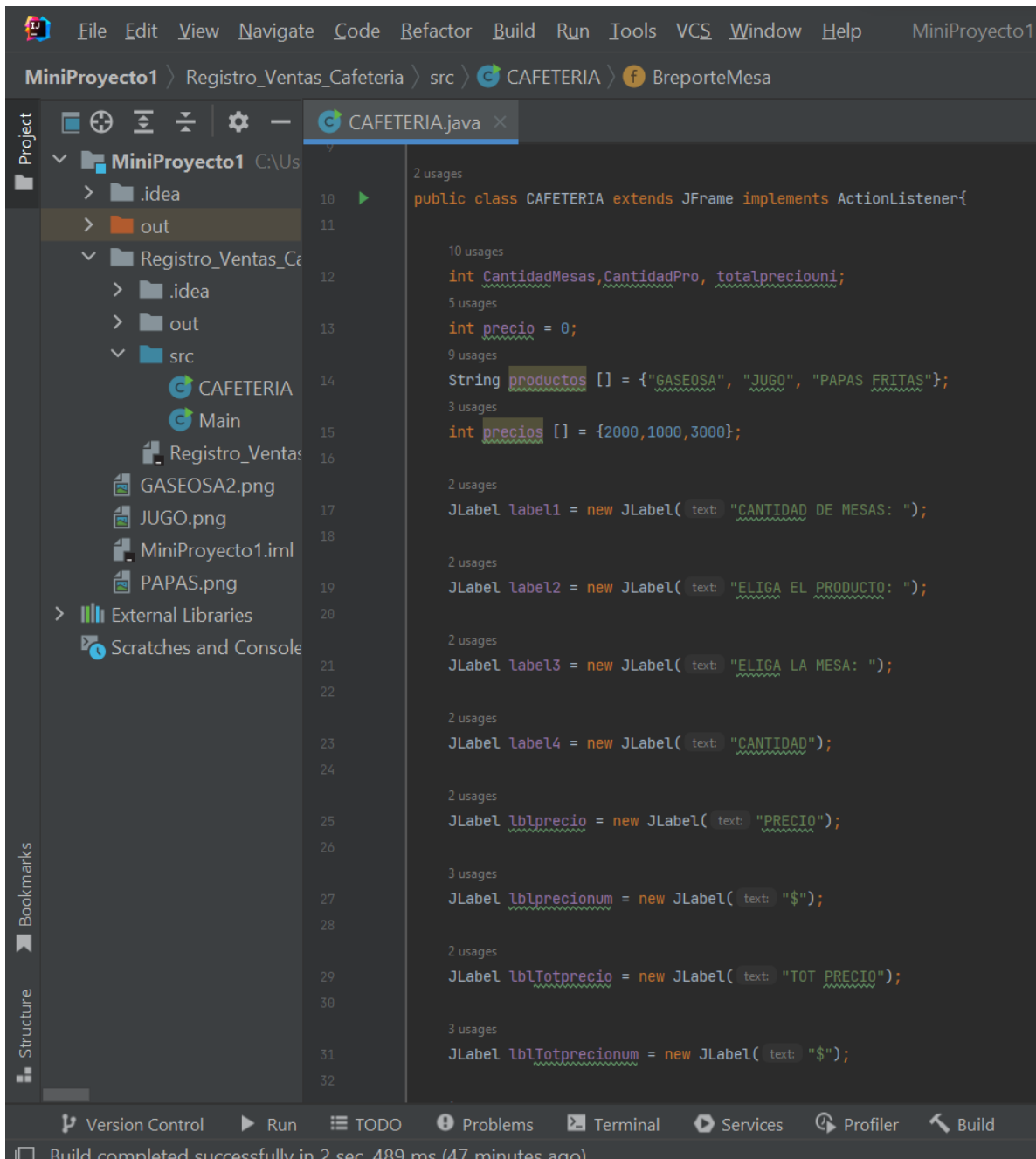
## DIAGRAMA UML:

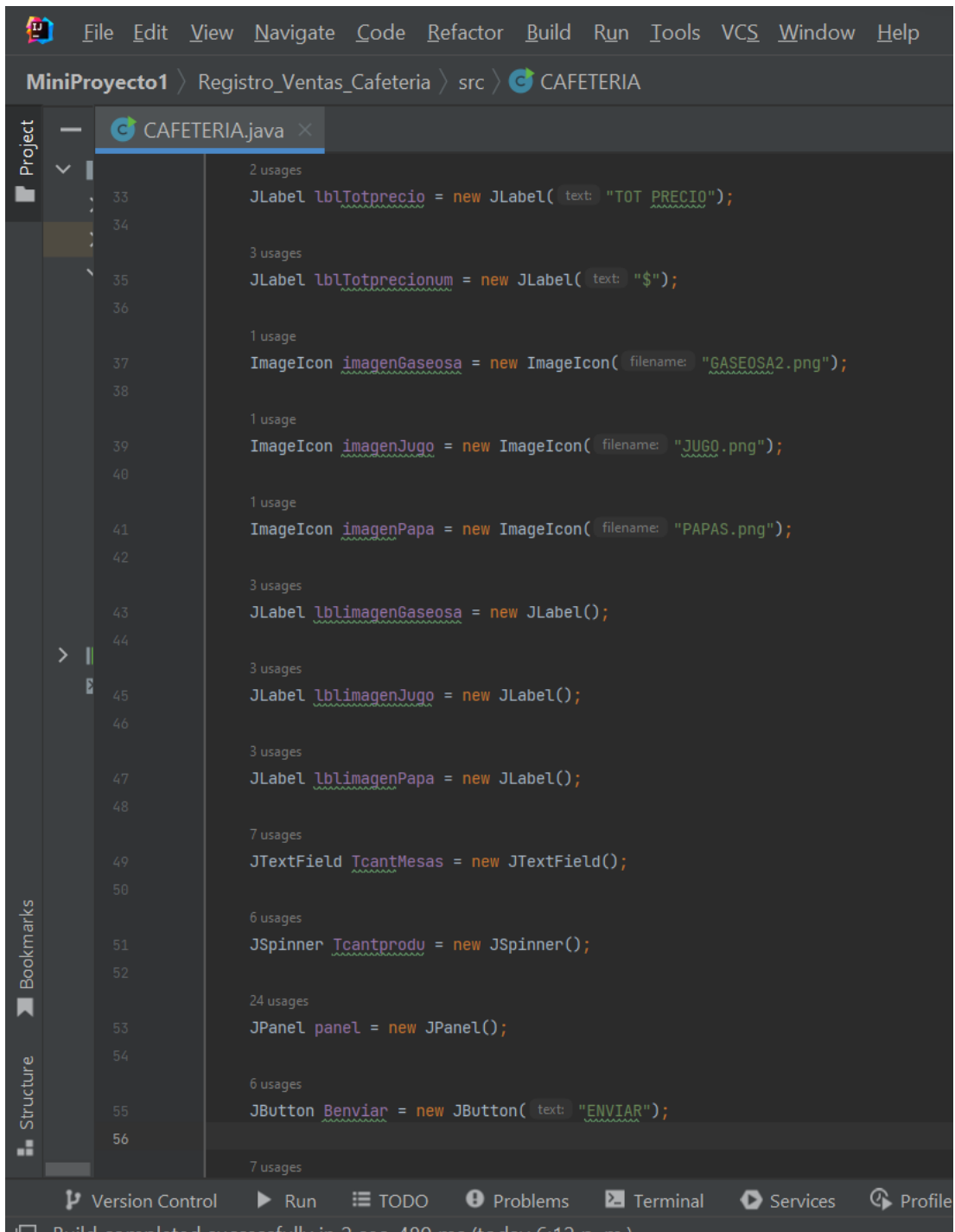


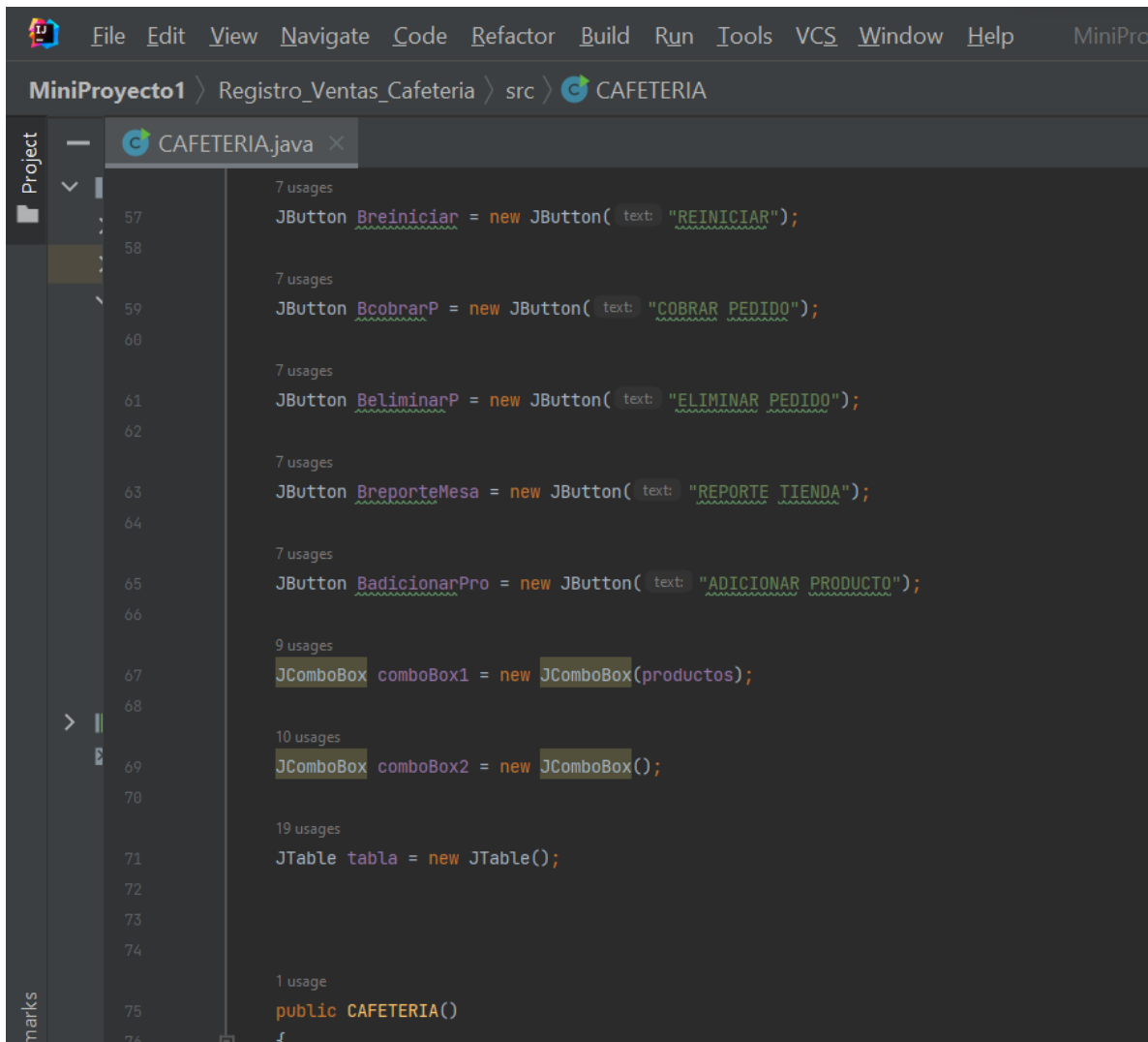
De comienzo se crea la clase CAFETERIA y se le agrega el JFrame como también ActionListener, los cuales permite crear la interfaz y agregarle eventos a elemento agregados a la ventana.

En esta primera parte también se crean las variables que guardaran información importante para el funcionamiento de la interfaz, se crean ocho Labels, tres ImageIcon las cuales son agregadas en un JLabel respectivamente. Se crea en un JTextField el cual guardara el numero de mesas que tiene disponible la tienda.

También se crean seis botones y dos JComboBox el primero muestra y permite seleccionar los productos disponibles y el segundo muestra y permite seleccionar las mesas.

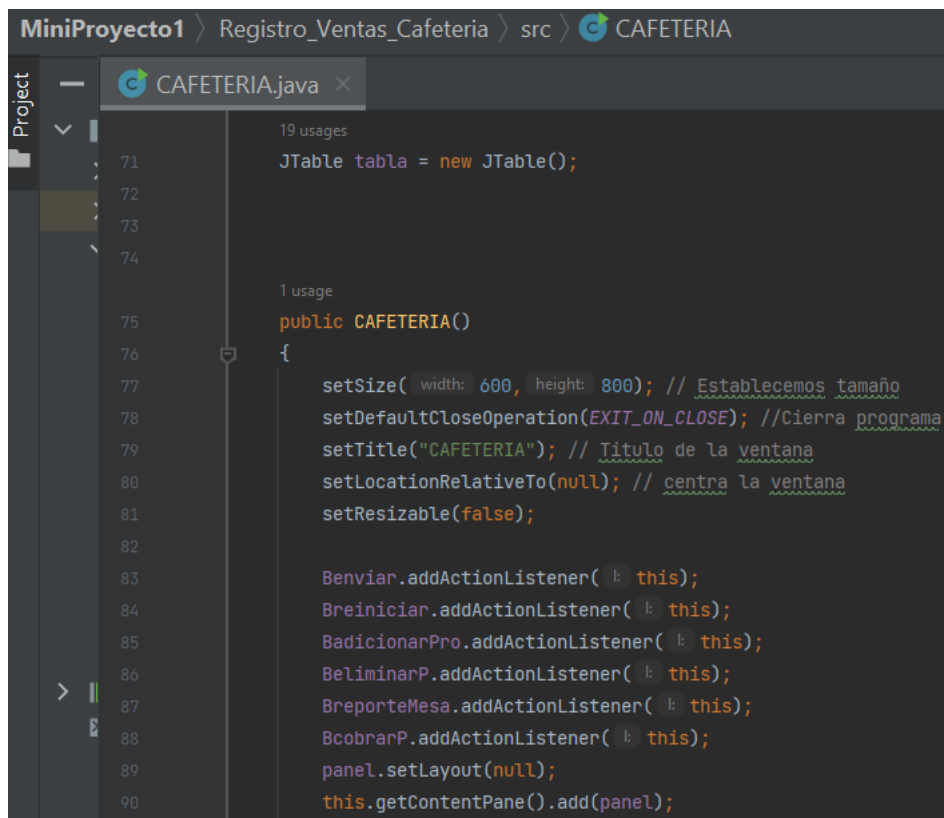






```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help MiniPro
MiniProyecto1 > Registro_Ventas_Cafeteria > src > CAFETERIA
CAFETERIA.java x
Project
57 JButton Breiniciar = new JButton( text: "REINICIAR");
58
59 JButton BcobrarP = new JButton( text: "COBRAR PEDIDO");
60
61 JButton BeliminarP = new JButton( text: "ELIMINAR PEDIDO");
62
63 JButton BreporteMesa = new JButton( text: "REPORTE TIENDA");
64
65 JButton BadicionarPro = new JButton( text: "ADICIONAR PRODUCTO");
66
67 JComboBox comboBox1 = new JComboBox(productos);
68
69 JComboBox comboBox2 = new JComboBox();
70
71 JTable tabla = new JTable();
72
73
74
75 public CAFETERIA()
76 {
```

Luego se crea el constructor de la clase en donde se define los parámetros y características de la ventana, además se adicionan los eventos de los botones y se desactiva el diseño de panel para que se pueda editar.

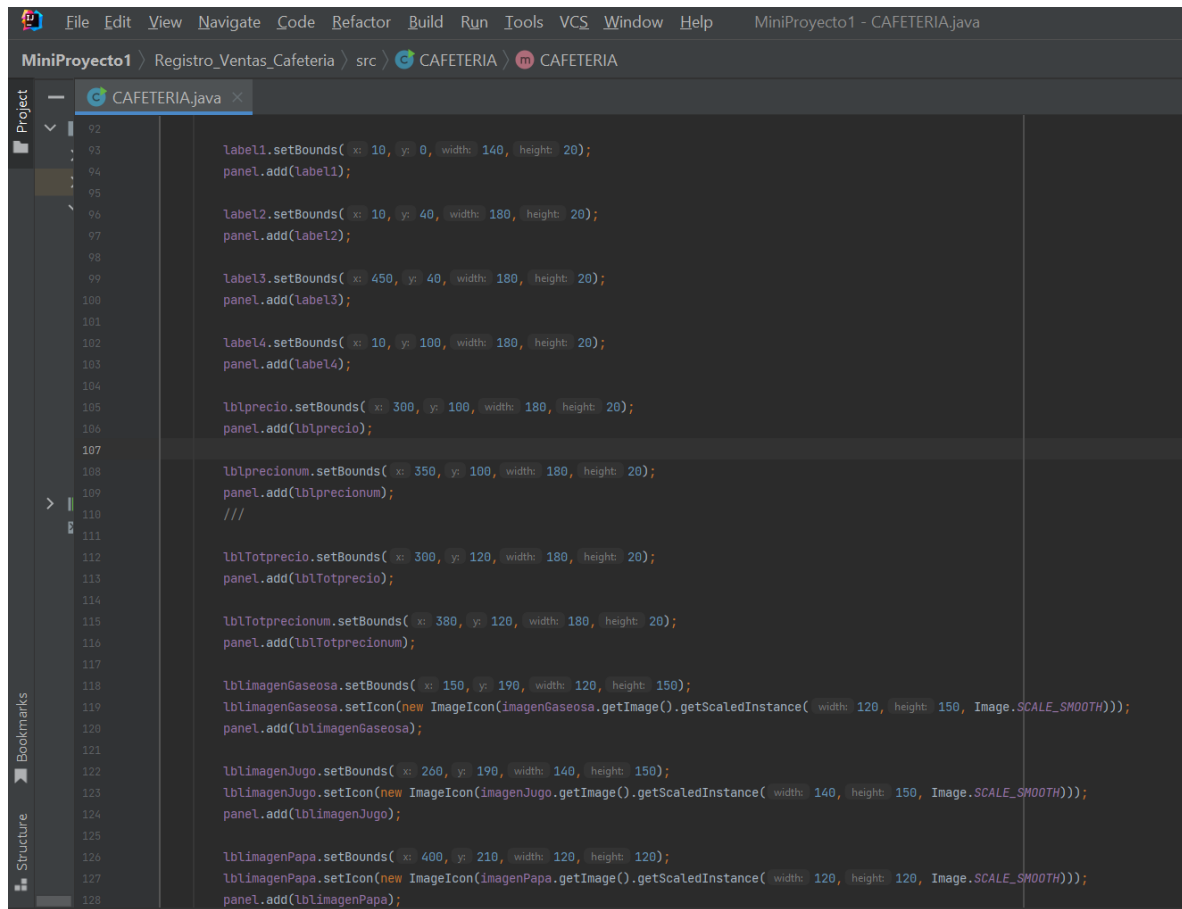


```
MiniProyecto1 > Registro_Ventas_Cafeteria > src > CAFETERIA
CAFETERIA.java x
19 usages
JTable tabla = new JTable();

1 usage
public CAFETERIA()
{
    setSize( width: 600, height: 800); // Establecemos tamaño
    setDefaultCloseOperation(EXIT_ON_CLOSE); //Cierra programa
    setTitle("CAFETERIA"); // Titulo de la ventana
    setLocationRelativeTo(null); // centra la ventana
    setResizable(false);

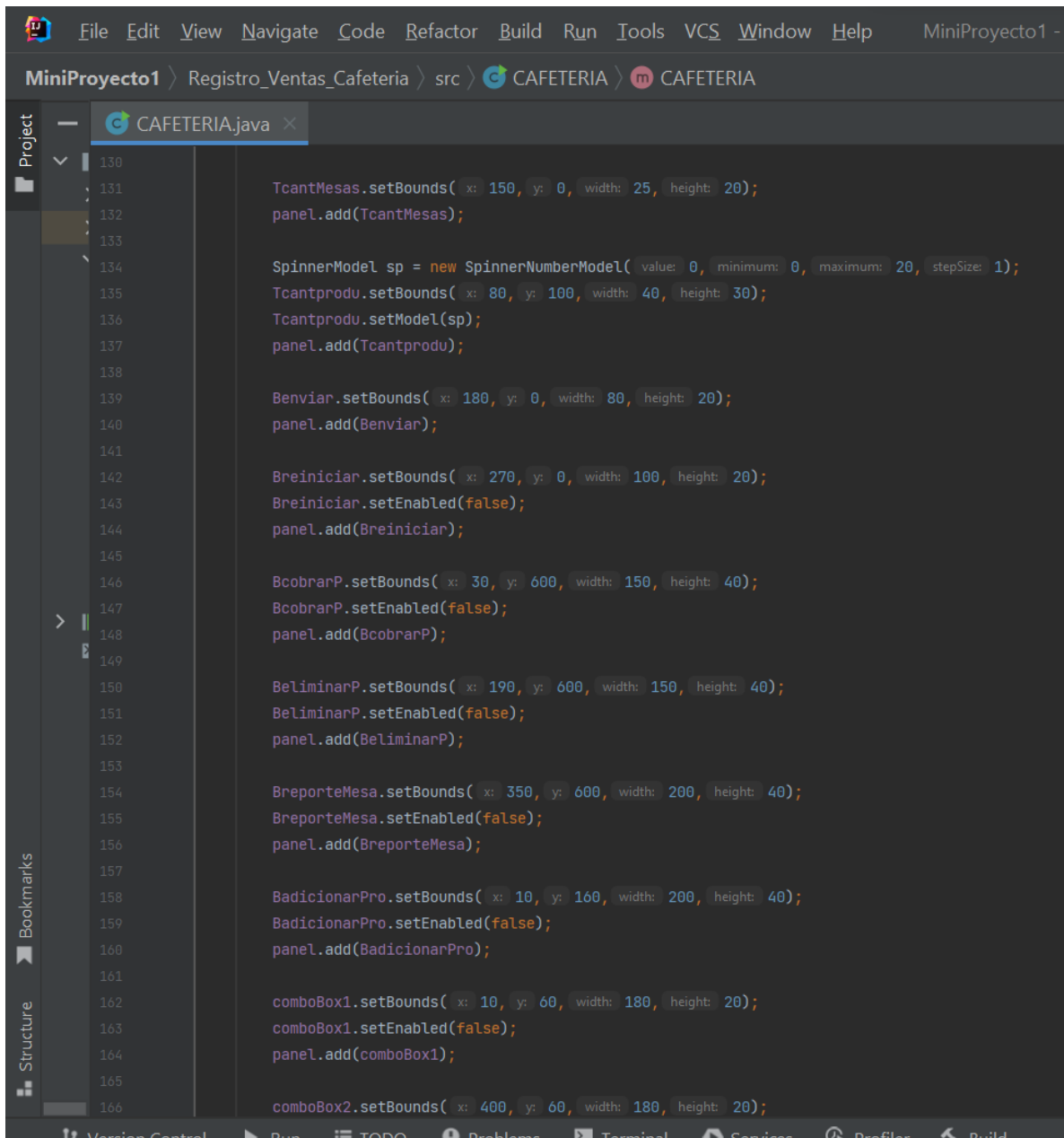
    Benviar.addActionListener( e: this);
    Breiniciar.addActionListener( e: this);
    BadicionarPro.addActionListener( e: this);
    BeliminarP.addActionListener( e: this);
    BreporteMesa.addActionListener( e: this);
    BcobrarP.addActionListener( e: this);
    panel.setLayout(null);
    this.getContentPane().add(panel);
}
```

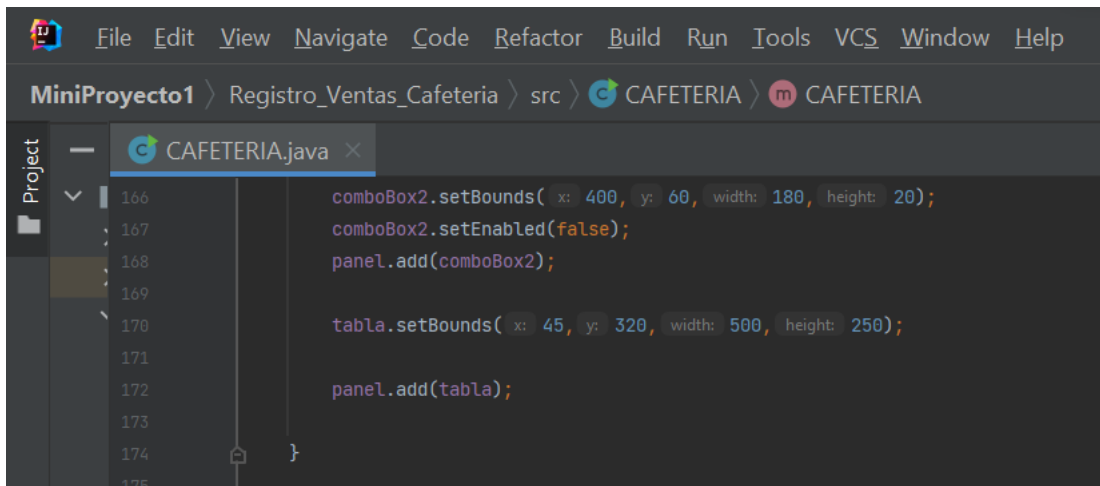
Se define tanto el tamaño como la posición de los JLabel, JButton; JComboBox, JTextField, JSpinner y la JTabla y se agregan al panel para que así aparezcan en la ventana.



```
92
93     label1.setBounds( x: 10, y: 0, width: 140, height: 20);
94     panel.add(label1);
95
96     label2.setBounds( x: 10, y: 40, width: 180, height: 20);
97     panel.add(label2);
98
99     label3.setBounds( x: 450, y: 40, width: 180, height: 20);
100    panel.add(label3);
101
102    label4.setBounds( x: 10, y: 100, width: 180, height: 20);
103    panel.add(label4);
104
105    lblprecio.setBounds( x: 300, y: 100, width: 180, height: 20);
106    panel.add(lblprecio);
107
108    lblprecionum.setBounds( x: 350, y: 100, width: 180, height: 20);
109    panel.add(lblprecionum);
110    ///
111
112    lblTotprecio.setBounds( x: 300, y: 120, width: 180, height: 20);
113    panel.add(lblTotprecio);
114
115    lblTotprecionum.setBounds( x: 380, y: 120, width: 180, height: 20);
116    panel.add(lblTotprecionum);
117
118    lblImagenGaseosa.setBounds( x: 150, y: 190, width: 120, height: 150);
119    lblImagenGaseosa.setIcon(new ImageIcon(imagenGaseosa.getImage().getScaledInstance( width: 120, height: 150, Image.SCALE_SMOOTH)));
120    panel.add(lblImagenGaseosa);
121
122    lblImagenJugo.setBounds( x: 260, y: 190, width: 140, height: 150);
123    lblImagenJugo.setIcon(new ImageIcon(imagenJugo.getImage().getScaledInstance( width: 140, height: 150, Image.SCALE_SMOOTH)));
124    panel.add(lblImagenJugo);
125
126    lblImagenPapa.setBounds( x: 400, y: 210, width: 120, height: 120);
127    lblImagenPapa.setIcon(new ImageIcon(imagenPapa.getImage().getScaledInstance( width: 120, height: 120, Image.SCALE_SMOOTH)));
128    panel.add(lblImagenPapa);
```

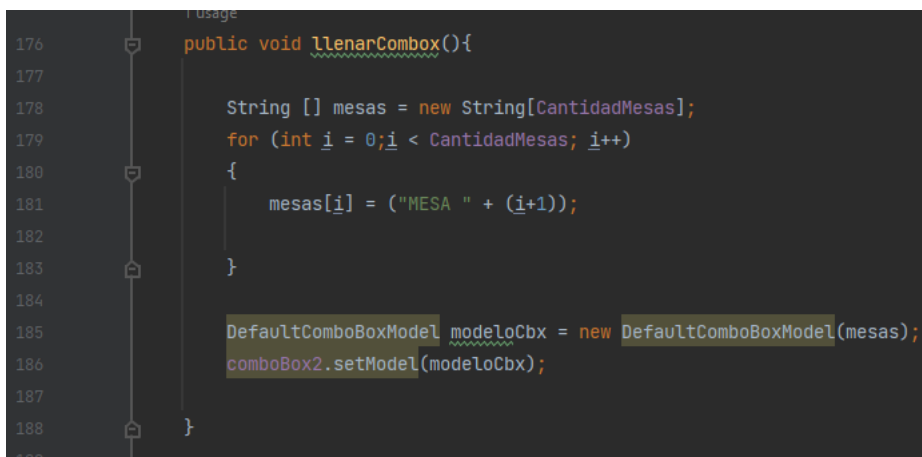






```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
MiniProyecto1 > Registro_Ventas_Cafeteria > src > CAFETERIA > CAFETERIA
Project
CAFETERIA.java x
166 comboBox2.setBounds( x: 400, y: 60, width: 180, height: 20);
167 comboBox2.setEnabled(false);
168 panel.add(comboBox2);
169
170 tabla.setBounds( x: 45, y: 320, width: 500, height: 250);
171
172 panel.add(tabla);
173
174 }
175
```

Después de terminar el código de el constructor se crea un método que como su nombre lo dice sirve para llenar el comboBox2, aquí se crea un arreglo de Strings, el cual su tamaño depende de la cantidad de mesas que ingrese el usuario. Se crea un for, el cual crea y guarda en el arreglo los nombres de las mesas. Luego de que el arreglo esta lleno se pasa al comboBox2 por medio del modelo, el cual es llenado con los nombres de las mesas. Cabe resaltar que ese método no se ejecuta hasta que el botón Benviar es oprimido.



```
Usage
176 public void llenarCombox(){
177
178     String [] mesas = new String[CantidadMesas];
179     for (int i = 0; i < CantidadMesas; i++)
180     {
181         mesas[i] = ("MESA " + (i+1));
182     }
183
184
185     DefaultComboBoxModel modeloCbx = new DefaultComboBoxModel(mesas);
186     comboBox2.setModel(modeloCbx);
187
188 }
189
```

Luego se crea otro método que permite llenar la JTable, como primer paso se crea un arreglo bidimensional con los datos que queremos mostrar en la tabla, como segundo paso se crea un for anidado, en el primero se agrega el

número de mesa en la primera columna y en el segundo se inicializan la cantidad de productos en 0, por ultimo se asigna a la tabla por medio de su modelo.

```
190 public void llenartabla()  
191 {  
192     Object[][] datos = new Object[CantidadMesas][productos.length + 1];  
193  
194     for (int i = 0; i < CantidadMesas; i++) {  
195         // Agregamos el número de mesa en la primera columna  
196         datos[i][0] = "MESA " + (i+1);  
197         // Agregamos la cantidad de cada producto en las columnas siguientes  
198         for (int j = 0; j < productos.length; j++) {  
199             datos[i][j+1] = 0; // Inicializamos la cantidad en cero  
200         }  
201     }  
202  
203     // Creamos el modelo de tabla con el arreglo de datos y los nombres de las columnas  
204     DefaultTableModel modeloTabla = new DefaultTableModel(datos,  
205         new String[]{"MESA", "GASEOSA", "JUGO", "PAPAS FRITAS"});  
206     tabla.setModel(modeloTabla);  
207  
208  
209
```

Por ultimo y antes de llamar el método main para que se ejecute el programa se crea el método actionPerformed el cual permite crear eventos en la interfaz de la tienda, de principio le damos un evento al botón Benviar, el cual luego de comprobar que lo que sea ingresado en el JTextField es un numero entero procede a activar el botón BadicionarPro y los dos comboBox, pero además de esto ese numero que se digito es guardado en la variable CantidadMesas y además ejecuta los métodos llenarCombox y llenarTabla los cuales dependiendo de la cantidad de mesas se llenan.

```
209 |
210 | @
211 | {
212 |     JButton fb = (JButton) e.getSource();
213 |
214 |     if (fb == Benviar){
215 |         try {
216 |             BadicionarPro.setEnabled(true);
217 |             comboBox1.setEnabled(true);
218 |             comboBox2.setEnabled(true);
219 |             CantidadMesas = Integer.parseInt(TcantMesas.getText());
220 |             Benviar.setEnabled(false);
221 |             TcantMesas.setEnabled(false);
222 |             Breiniciar.setEnabled(true);
223 |             llenarCombox();
224 |             llenartabla();
225 |
226 |         } catch (NumberFormatException nfe)
227 |         {
228 |             JOptionPane.showMessageDialog( parentComponent: null, message: "POR " +
229 |                 "FAVOR DIGITE UN NUMERO");
230 |         }
231 |
232 |     }
```

Al evento del botón Breiniciar permite poder eliminar y volver a iniciar la cantidad de mesas disponibles y por ende se les restringe la acción a los botones y comboBox que no tendría sentido que estuvieran activados, de acuerdo con la lógica del programa.

```
232 |     }
233 |     if (fb == Breiniciar){
234 |
235 |         comboBox1.setEnabled(true);
236 |         comboBox2.setEnabled(true);
237 |         BadicionarPro.setEnabled(false);
238 |         comboBox1.setEnabled(false);
239 |         comboBox2.setEnabled(false);
240 |         BcobrarP.setEnabled(false);
241 |         BreporteMesa.setEnabled(false);
242 |         BeliminarP.setEnabled(false);
243 |         CantidadMesas = 0;
244 |         CantidadMesas = Integer.parseInt(TcantMesas.getText());
245 |         Benviar.setEnabled(true);
246 |         TcantMesas.setEnabled(true);
247 |         Breiniciar.setEnabled(false);
248 |         TcantMesas.setText("");
249 |         comboBox2.removeAllItems();
250 |     }
```

Al tercer botón BcobrarP, el cual me va permitir que al momento de seleccionar una fila (mesa) en la tabla muestre los productos adquiridos por esa mesa y el total a pagar, esto se hace de acuerdo con unas instrucciones, pues primero se guarda en una variable la posición de la fila seleccionada en la tabla, después se guarda la cantidad de productos consumidos por la mesa por cada producto, luego de obtener esa cantidad se realiza la multiplicación de acuerdo con el precio de cada producto como también el total a pagar, después de hacer todo lo anterior se crea una ventana la cual muestra toda la información extraída y por ultimo se borran los datos de la mesa para que pueda tener un nuevo pedido.

```
262         if (fb == BcobrarP)
263         {
264             int filaSeleccionada = tabla.getSelectedRow();
265             String mesaSelecion = tabla.getValueAt(filaSeleccionada, column: 0).toString();
266             int cantidadProducto1 = Integer.parseInt(tabla.getValueAt(filaSeleccionada, column: 1).toString());
267             int cantidadProducto2 = Integer.parseInt(tabla.getValueAt(filaSeleccionada, column: 2).toString());
268             int cantidadProducto3 = Integer.parseInt(tabla.getValueAt(filaSeleccionada, column: 3).toString());
269
270             int PagarGaseosa = cantidadProducto1 * 2000;
271             int PagarJugo = cantidadProducto2 * 1000;
272             int PagarPapa = cantidadProducto3 * 3000;
273             int totalPagar = (PagarGaseosa + PagarJugo + PagarPapa);
274
275             if (cantidadProducto1 == 0 & cantidadProducto2 == 0 & cantidadProducto3 == 0 )
276             {
277                 JOptionPane.showMessageDialog( parentComponent: null, message: ";ERROR!" + "\n" + "La mesa seleccionada no tiene una orden");
278             }
279             else
280             {
281                 JOptionPane.showMessageDialog( parentComponent: null, message: "MESA: " + mesaSelecion + "\n" +
282                     "Cantidad Gaseosa:" + cantidadProducto1 + " $" + PagarGaseosa + "\n" +
283                     "Cantidad Jugo:" + cantidadProducto2 + " $" + PagarJugo + "\n" +
284                     "Cantidad Papas Fritas:" + cantidadProducto3 + " $" + PagarPapa + "\n" +
285                     "TOTAL A PAGAR: $" + totalPagar );
286             }
287
288             int row = tabla.getSelectedRow();
289             tabla.setValueAt( aValue: 0, row, column: 1);
290             tabla.setValueAt( aValue: 0, row, column: 2);
291             tabla.setValueAt( aValue: 0, row, column: 3);
292         }
293     }
```

EL botón BadicionarPro permite que al momento de presionarlo tome el producto seleccionado en el comboBox1, la cantidad seleccionada en el Spinner y la posición de la mesa seleccionada, para así imprimir en la tabla los datos anteriores.

```

284
285     if (fb == BadicionarPro)
286     {
287         BcobrarP.setEnabled(true);
288         BreporteMesa.setEnabled(true);
289         BeliminarP.setEnabled(true);
290         int index = comboBox1.getSelectedIndex();
291         productoSel = productos[index];
292         mesaSel = comboBox2.getSelectedItem().toString();
293         posicionMesa = comboBox2.getSelectedIndex();
294         int cantidadProd = (int) Tcantprodu.getValue();
295         int precioProd = precios[index];
296         int totalPrecio = cantidadProd * precioProd;
297         tabla.setValueAt(cantidadProd, posicionMesa, column: index+1);
298         precio = precio + totalPrecio;
299     }
300

```

Al oprimir el botón BeliminarP elimina los datos que están ingresados de la mesa que esta seleccionada en la tabla.

```

if (fb == BeliminarP){
    int row = tabla.getSelectedRow();
    tabla.setValueAt(aValue: 0, row, column: 1);
    tabla.setValueAt(aValue: 0, row, column: 2);
    tabla.setValueAt(aValue: 0, row, column: 3);
}

```

Por último en cuanto a los eventos de los botones está el botón BreporteMesa, el cual al ser accionado crea una variable que guarda el total de ventas la cual es definida por dos for anidados que básicamente lo que hacen es que recorre la tabla y toma la cantidad de productos comprados por cada mesa y de acuerdo con esta se multiplica con su precio correspondiente definidos en el arreglo de precios.

```

if (fb == BreporteMesa)
{
    // Obtener el total de ventas
    int totalVentas = 0;
    for (int i = 0; i < CantidadMesas; i++) {
        for (int j = 0; j < productos.length; j++) {
            int cantidad = Integer.parseInt(tabla.getValueAt(i, column: j+1).toString());
            totalVentas += cantidad * precios[j];
        }
    }
}

```

Como segunda acción se guarda la división entre el total de ventas y la cantidad de mesas lo que básicamente daría como resultado el promedio de ventas por mesa.

```
319 // Calcular los ingresos promedio por mesa
320 double ingresosPromedio = (double) totalVentas / CantidadMesas;
321
```

Como ultima acción este botón ayuda encontrar el producto mas vendido, lo cual se logra creando un arreglo de enteros llamado cantidades y creando dos for anidados los cuales recorren las filas de los productos de cada mesa y tomando su valor para sumarlo al arreglo, luego de esto se crea otra variable llamada masVendido donde se define dentro de un for y un if y permite dar como resultado del producto mas vendido. Por ultimo se imprime todos los datos.

```

307
308     if (fb == BreporteMesa)
309     {
310         // Obtener el total de ventas
311         int totalVentas = 0;
312         for (int i = 0; i < CantidadMesas; i++) {
313             for (int j = 0; j < productos.length; j++) {
314                 int cantidad = Integer.parseInt(tabla.getValueAt(i, columna: j+1).toString());
315                 totalVentas += cantidad * precios[j];
316             }
317         }
318
319         // Calcular los ingresos promedio por mesa
320         double ingresosPromedio = (double) totalVentas / CantidadMesas;
321
322         // Encontrar el producto más vendido
323         int[] cantidades = new int[productos.length];
324         for (int i = 0; i < CantidadMesas; i++) {
325             for (int j = 0; j < productos.length; j++) {
326                 int cantidad = Integer.parseInt(tabla.getValueAt(i, columna: j+1).toString());
327                 cantidades[j] += cantidad;
328             }
329         }
330         int masVendido = 0;
331         for (int i = 1; i < productos.length; i++) {
332             if (cantidades[i] > cantidades[masVendido]) {
333                 masVendido = i;
334             }
335         }
336         // Mostrar la información en pantalla
337         String mensaje = "Total de ventas: $" + totalVentas + "\n";
338         mensaje += "Ingresos promedio por mesa: $" + String.format("%.2f", ingresosPromedio) + "\n";
339         mensaje += "Producto más vendido: " + productos[masVendido] + "\n";
340         JOptionPane.showMessageDialog( parentComponent: null, mensaje);
341
342     }

```

Por último, en el programa le doy una acción tanto cuando se elige un elemento de el combobox1 como la cantidad de Tcantproducto es decir el Spinner. Al momento de seleccionar el producto y la cantidad del mismo en el JLabel lblprecionum se visualiza el precio del producto por unida y al momento de elegir la cantidad del JLabel lblTotprecionum muestra el precio del producto, pero de acuerdo con su cantidad.

Para le ejecución se ejecuta la clase CAFETERIA en el método main.



```

comboBox1.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {

        precio = precios[comboBox1.getSelectedIndex()];
        lblprecionum.setText("$"+precio);

    }
});

Tcantprodu.addChangeListener(new ChangeListener() {
    @Override
    public void stateChanged(ChangeEvent e) {
        CantidadPro = Integer.parseInt(Tcantprodu.getValue().toString());
        totalpreciouni = precio*CantidadPro;
        lblTotprecionum.setText("$"+totalpreciouni);
    }
});

}

no usages
public static void main(String[] args)
{
    CAFETERIA cafeteria = new CAFETERIA();
    cafeteria.setVisible(true);
}

```

EJECUCION:

Programa Iniciado:

CAFETERIA

—

□

×

CANTIDAD DE MESAS:

ENVIAR

REINICIAR

ELIGA EL PRODUCTO:

GASEOSA

ELIGA LA MESA:




CANTIDAD

0

PRECIO \$

TOT PRECIO \$

ADICIONAR PRODUCTO



COBRAR PEDIDO

ELIMINAR PEDIDO

REPORTE TIENDA

Se ingresa la cantidad de mesas(10):

CAFETERIA

CANTIDAD DE MESAS: 10

ENVIAR

REINICIAR




ELIGA EL PRODUCTO:  
GASEOSA

ELIGA LA MESA:  
MESA 1

CANTIDAD 0

PRECIO \$  
TOT PRECIO \$

ADICIONAR PRODUCTO



MESA 1	0	0	0
MESA 2	0	0	0
MESA 3	0	0	0
MESA 4	0	0	0
MESA 5	0	0	0
MESA 6	0	0	0
MESA 7	0	0	0
MESA 8	0	0	0
MESA 9	0	0	0
MESA 10	0	0	0

COBRAR PEDIDO

ELIMINAR PEDIDO

REPORTE TIENDA

Se adiciona los productos de la mesa 1:

CAFETERIA

CANTIDAD DE MESAS: 10

ENVIAR

REINICIAR

ELIGA EL PRODUCTO:

PAPAS FRITAS

ELIGA LA MESA:

MESA 1




CANTIDAD

2

PRECIO \$3000

TOT PRECIO \$6000

ADICIONAR PRODUCTO



MESA 1	2	2	2
MESA 2	0	0	0
MESA 3	0	0	0
MESA 4	0	0	0
MESA 5	0	0	0
MESA 6	0	0	0
MESA 7	0	0	0
MESA 8	0	0	0
MESA 9	0	0	0
MESA 10	0	0	0

COBRAR PEDIDO

ELIMINAR PEDIDO

REPORTE TIENDA

Se adiciona los productos de la mesa 3:

CAFETERIA

CANTIDAD DE MESAS: 10

ENVIAR

REINICIAR




ELIGA EL PRODUCTO:  
GASEOSA

ELIGA LA MESA:  
MESA 3

CANTIDAD 1

PRECIO \$2000  
TOT PRECIO \$2000

ADICIONAR PRODUCTO



MESA 1	2	2	2
MESA 2	0	0	0
MESA 3	1	0	3
MESA 4	0	0	0
MESA 5	0	0	0
MESA 6	0	0	0
MESA 7	0	0	0
MESA 8	0	0	0
MESA 9	0	0	0
MESA 10	0	0	0

COBRAR PEDIDO

ELIMINAR PEDIDO

REPORTE TIENDA

Se cobra pedido de la mesa 1:

CAFETERIA

CANTIDAD DE MESAS: 10

ENVIAR

REINICIAR

ELIGA EL PRODUCTO:

GASEOSA

ELIGA LA MESA:

MESA 2


CANTIDAD

1

PRECIO \$2000

TOT PRECIO \$2000

ADICIONAR PRODUCTO



Message

i

MESA: MESA 1

Cantidad Gaseosa:2 \$4000

Cantidad Jugo:2 \$2000

Cantidad Papas Fritas:2 \$6000

TOTAL A PAGAR: \$12000

OK

MESA 1	2	2	2
MESA 2	1	3	4
MESA 3	1	0	3
MESA 4	7	0	7
MESA 5	3	3	3
MESA 6	1	1	1
MESA 7	3	4	7
MESA 8	10	12	5
MESA 9	0	1	1
MESA 10	2	1	2

COBRAR PEDIDO

ELIMINAR PEDIDO

REPORTE TIENDA

Se cobra pedido de la mesa 2:

CAFETERIA

CANTIDAD DE MESAS: 10

ENVIAR

REINICIAR

ELIGA EL PRODUCTO:

GASEOSA

ELIGA LA MESA:

MESA 2


CANTIDAD

1

PRECIO \$2000

TOT PRECIO \$2000

ADICIONAR PRODUCTO



Message

i

MESA: MESA 2

Cantidad Gaseosa:1 \$2000

Cantidad Jugo:3 \$3000

Cantidad Papas Fritas:4 \$12000

TOTAL A PAGAR: \$17000

OK


MESA 1	0	0	0
MESA 2	1	3	4
MESA 3	1	0	3
MESA 4	7	0	7
MESA 5	3	3	3
MESA 6	1	1	1
MESA 7	3	4	7
MESA 8	10	12	5
MESA 9	0	1	1
MESA 10	2	1	2

COBRAR PEDIDO

ELIMINAR PEDIDO

REPORTE TIENDA

Se elimina el pedido de la mesa 8:

 CAFETERIA

CANTIDAD DE MESAS:

ENVIAR

REINICIAR

ELIGA EL PRODUCTO:  

GASEOSA

ELIGA LA MESA:  




MESA 2

CANTIDAD 

1

PRECIO \$2000  
TOT PRECIO \$2000

ADICIONAR PRODUCTO



MESA 1	0	0	0
MESA 2	0	0	0
MESA 3	1	0	3
MESA 4	7	0	7
MESA 5	3	3	3
MESA 6	1	1	1
MESA 7	3	4	7
MESA 8	0	0	0
MESA 9	0	1	1
MESA 10	2	1	2

COBRAR PEDIDO

ELIMINAR PEDIDO

REPORTE TIENDA



Se obtiene el reporte de las ventas de la tienda:

CAFETERIA


CANTIDAD DE MESAS: 10

ELIGA EL PRODUCTO: GASEOSA 

ELIGA LA MESA: MESA 2

CANTIDAD 1 

PRECIO \$2000  
TOT PRECIO \$2000



Message

i

Total de ventas: \$116000  
Ingresos promedio por mesa: \$11600,00  
Producto más vendido: PAPAS FRITAS

MESA 1	0	0	0
MESA 2	0	0	0
MESA 3	1	0	3
MESA 4	7	0	7
MESA 5	3	3	3
MESA 6	1	1	1
MESA 7	3	4	7
MESA 8	0	0	0
MESA 9	0	1	1
MESA 10	2	1	2