

# Informe ProyectoPOO

**Por:** Daniel Medina

**Fecha:** 19/03/2020

**Tema:** Creación de un programa con interfaz gráficas y venta de productos.

## Ejercicio 1

### Definición del problema

Realizar un punto de venta, aplicando POO, calculando el precio del producto por la cantidad solicitada, y el total a pagar.

### Análisis

#### Entrada

Producto, cantidad

#### Proceso

Subtotal = cantidad \* precioP

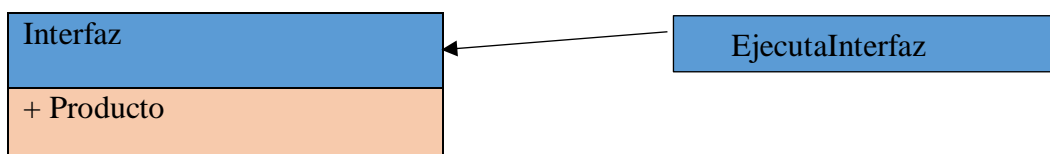
iva = sub +0.12

total = sub + iva

#### Salida

Imprime “Producto, cantidad, subtotal, iva, total”

### Diagrama de clases



+ cantidad
+ precioP
+ total
- void formWindowOpened
- void prodcomestibles
- void textprecio
- void agregar
- void texttotal
- void listaproducto
- void eliminar
- void listaproductos
- void salir
- <u>void jButton</u>
- <u>void subtotal</u>
- void subtotal
- void jtotal
- void lista
- <u>void sumatotal</u>

## Pseudocodigo

### Clase interfaz

1. Declaración de variables

precioP real

cant real

sub real

elemento entero

## 2. Creación de objetos

```
DecimalFormat d = new DecimalFormat("0.00")
```

```
SpinnerNumberModel s = new SpinnerNumberModel()
```

```
DefaultListModel listpro = new DefaultListModel()
```

```
DefaultListModel listpre = new DefaultListModel()
```

```
DefaultListModel listcant = new DefaultListModel()
```

```
DefaultListModel listsub = new DefaultListModel()
```

```
DefaultListModel listiva = new DefaultListModel()
```

```
DefaultListModel listtotal = new DefaultListModel()
```

## 3. Método de inicializarcomponentes ()

```
initComponents()
```

```
s.setMaximum(20)
```

```
s.setMinimum(0)
```

```
listaproductos.setModel(listpro)
```

```
listaprecios.setModel(listpre)
```

```
listacatidad.setModel(listcant)
```

```
listasubtotal.setModel(listsub)
```

```
iva.setModel(listiva)
```

```
jtotal.setModel(listtotal)

texttotal.setEditable(true)

textprecio.setEditable(false)

subtotal.setEditable(false)
```

Fin método inicializarcomponentes

#### 4. Metodo formWindowOpened()

```
prodcomestibles.agregar("Arroz");

prodcomestibles.agregar ("Tomate");

prodcomestibles. agregar ("Salmon");

prodcomestibles. agregar ("Salas Tomate");

prodcomestibles. agregar ("Mayonesa");
```

Fin método formWindowOpened

#### 5. Método prodcomestibles()

Switch

Caso “Arroz”

Precio = 0.60

Break;

Caso “Tomate”

Precio = 0.20

Break;

Caso "Salmon"

Precio = 1.25

Break;

Case "Salas Tomate"

Precio = 0.80

Break;

Caso "Mayonesa"

Precio = 0.80

Break;

Fin método prodcomestibles

6. Método textprecio()

Return textprecio

Fin método textprecio

7. Método cantidad()

cant = real.cantidad.obtenervvalor

subtotal = (precioP \*cant)

imprime subtotal.establecertexto

Fin método cantidad

8. Metodo agregarboton()

Cadena cc = prodcomestibles.seleccionarelemento

Para entero i = 0, i < listaproductos, i++

Si listaproductos.obtenerelemento

Opcionpane(El producto ya existe)

Retunr

Fin si

Fin para

tt real

ivaE real

Cadena produc = prodcomestibles. Seleccionaritem

Entero c = paseentero(cantidad)

elemento += 1

lista(produc, precioP, c, sub, ivaE, tt)

sumatotal()

Fin método agregarboton

9. Metodo texttotal()

Return texttotal

Fin método agregarboton

10. Metodo listaproductos()

Return listaproductos

Fin método listaproductos

11. Metodo eliminar()

enterp n = listaproductos.obtenerseleccion();

Si (n > -1)

listpro.removeelemento(n);

listpre. removeelemento (n);

```
listcant. removeelemento (n);  
listsub. removeelemento (n);  
listiva. removeelemento (n);  
listtotal. removeelemento (n);  
elemento -= 1;  
  
fin si
```

Fin método eliminar

## 12. Método listaproductos()

```
entero n = listaproductos.obtenerseleccionar();  
  
Si (n == -1)  
  
sino  
  
listaproductos.establecerseleccion(n);  
listaprecios. establecerseleccion (n);  
listacatidad. establecerseleccion (n);  
listasubtotal. establecerseleccion (n);  
iva. establecerseleccion (n);  
jtotal. establecerseleccion (n);  
  
fin si
```

Fin método listaproductos

## 13. Método salir()

```
entero men = opcion.dialogo("Desea salir ")  
  
Si (men == opcion.Si )
```

Sistema.salir(0);

Fin Si

Fin método salir

#### 14. Método botónborrar

Entro men = opcion.dialogo("Desea ingresar nuevamente").YES\_NO\_OPTION)

Si (men == opcion.YES\_OPTION)

Real precioP = 0;

cant = 0;

sub = 0;

int elemento = 0

textprecio.establecerTexto("")

subtotal. establecerTexto ("")

texttotal. establecerTexto ("")

listpro.removeelementos();

listpre. removeelementos ();

listcant. removeelementos ();

listsub. removeelementos ();

listiva. removeelementos ();

listtotal. removeelementos ();

Fin si

Fin método botónborrar

#### 15. Método subtotal()

Return subtotal

Fin método subtotal

#### 16. Método jtotal ()



Return jtotal

Fin método jtotal

17. Metodo lista(cadena pro, real pre, entero cant, real sub, real v, real to)

listpro.agregarelemento(pro);

listpre. agregarelemento (d.format(pre));

listcant. agregarelemento (cant);

listsub. agregarelemento (d.format(sub));

listiva. agregarelemento (d.format(v));

listtotal. agregarelemento (to);

Fin método lista

18. Metodo sumatotal()

real n=0,

Real sumatt=0;

Para (entero i = 0, i <jtotal, i++ ){

n = Double.parseDouble(jtotal)

sumatt +=n

}

texttotal.establecertexto(formato(sumatt)+" \$");

Fin método sumatotal

19. Método main (ejecutainterfaz)

Nota: Se visualiza más código propio del JFrame el cual se agregan al importar sus paquetes.

```
java.awt.EventQueue.invokeLater(new Runnable() {  
    public void run() {  
        new interfaz().setVisible(true);  
    }  
});
```

Fin método main

## 20. Variables declaradas, pero no se pueden modificar

```
private javax.swing.JButton agregar;  
private javax.swing.JSpinner cantidad;  
private javax.swing.JButton eliminar;  
private javax.swing.JList<String> iva;  
private javax.swing.JButton jButton1;  
private javax.swing.JLabel jLabel1;  
private javax.swing.JLabel jLabel10;  
private javax.swing.JLabel jLabel11;  
private javax.swing.JLabel jLabel12;  
private javax.swing.JLabel jLabel13;  
private javax.swing.JLabel jLabel14;  
private javax.swing.JLabel jLabel15;  
private javax.swing.JLabel jLabel2;  
private javax.swing.JLabel jLabel3;  
private javax.swing.JLabel jLabel4;  
private javax.swing.JLabel jLabel5;
```

```
private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel9;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel3;

private javax.swing.JPanel jPanel4;

private javax.swing.JScrollPane jScrollPane3;

private javax.swing.JScrollPane jScrollPane4;

private javax.swing.JScrollPane jScrollPane5;

private javax.swing.JScrollPane jScrollPane6;

private javax.swing.JScrollPane jScrollPane7;

private javax.swing.JScrollPane jScrollPane8;

private javax.swing.JList<String> jtotal;

private javax.swing.JList<String> listacatidad;

private javax.swing.JList<String> listaprecios;

private javax.swing.JList<String> listaproductos;

private javax.swing.JList<String> listasubtotal;

private javax.swing.JComboBox<String> prodcomestibles;

private javax.swing.JTextField prubetotal;

private javax.swing.JButton salir;

private javax.swing.JTextField subtotal;

private javax.swing.JTextField texttotal;

private javax.swing.JTextField textprecio;
```

**Fin clase Interfaz**

## Funcionamiento de del programa y la interfaz

### Visualización de la interfaz.

PUNTO DE VENTA

**Venta de producto**

Productos: Arroz

Cantidad: 1

Precio: 0.60 \$

Subtotal: 0.60 \$

Agregar a la Lista

**Registro de productos**

Producto	Precio	Cantidad	Subtotal	Iva 12%	Total

Total a pagar:

Limpiar

Nuevo

Salir

En productos se muestran todos los elementos en stock de la tienda.

Venta de producto

Productos:

Salmon

Arroz

Tomate

Salmon

Salas Tomate

Mayonesa

Sifon

Salseras

Aliñadores

Cantidad:

1,25 \$

Subtotal:

1,25 \$

Agregar a la Lista

Registro de productos

Producto	Precio	Cantidad	Subtotal	Iva 12%	Total

Total a pagar:

Limpiar

Nuevo

Salir

Luego de seleccionar una Salmon se puede elegir, la cantidad, el precio y el subtotal.

**Venta de producto**

Productos: Salmon

Cantidad: 2

Precio: 1.25 \$

Subtotal: 2.50 \$

Agregar a la Lista

**Registro de productos**

Producto	Precio	Cantidad	Subtotal	Iva 12%	Total
Salmon	1,25	2	2,50	0,30	2.8

Total a pagar: 2.8

Limpiar

Nuevo

Salir

Se puede ver como en las listas de la interfaz de agregan el nombre del producto, su precio, cantidad, subtotal, iva y el total, además se ve el valor total a pagar.

**Venta de producto**

Productos: Salmon

Cantidad: 2 Precio: 1.25 \$ Subtotal: 2.50 \$

Agregar a la Lista

**Registro de productos**

Producto	Precio	Cantidad	Subtotal	Iva 12%	Total
Salmon	1,25	2	2,50	0,30	2.8

Total a pagar: 2.8

Limpiar

Nuevo

Salir

También podemos agregar más elementos a la lista y que se sumen.

**Venta de producto**

Productos:

Cantidad:  Precio:  Subtotal:

**Registro de productos**

Producto	Precio	Cantidad	Subtotal	Iva 12%	Total
Salmon	1,25	2	2,50	0,30	2,8
Tomate	0,20	2	0,40	0,05	0,448

Total a pagar:



Podemos eliminar un producto por medio del botón limpiar y seleccionado el producto.

**Venta de producto**

Productos: 

Tomate

Cantidad: 

2

Precio: 

0.20 \$

Subtotal: 

0.40 \$

Agregar a la Lista

**Registro de productos**

Producto	Precio	Cantidad	Subtotal	Iva 12%	Total
Salmon	1.25	2	2.50	0.30	2.8
Tomate	0.20	2	0.40	0.05	0.448

Total a pagar:

3.25

Limpiar

Nuevo

Salir

**Venta de producto**

Productos: 

Tomate

Cantidad: 

2

Precio: 

0.20 \$

Subtotal: 

0.40 \$

Agregar a la Lista

**Registro de productos**

Producto	Precio	Cantidad	Subtotal	Iva 12%	Total
Tomate	0.20	2	0.40	0.05	0.448

Total a pagar:

0.448

Limpiar

Nuevo

Salir

Mediante el botón Nuevo podemos reiniciar la interfaz, dejándola completamente limpia.

**Venta de producto**

Productos: Tomate

Cantidad: 2

Precio:

Subtotal:

Agregar a la Lista

**Registro de productos**

Producto	Precio	Cantidad	Subtotal	Iva 12%	Total
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

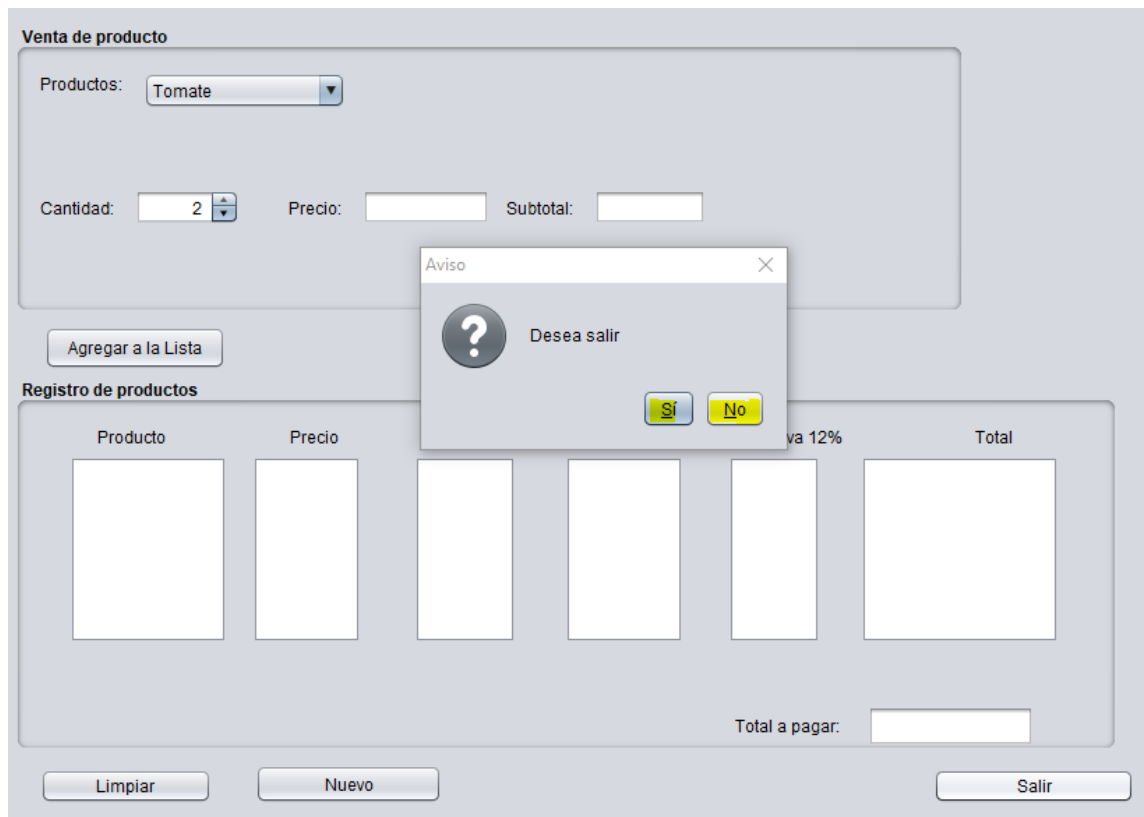
Total a pagar:

Limpiar

Nuevo

Salir

Por último, se puede salir y terminar el proceso mediante el botón salir el cual nos pide confirmar si o no, para salir del programa.



#### Recomendación:

- Al investigar el JFrame se puede encontrar miles de formas de utilizarlo, con el cual se confunde mucho usuario, por este motivo se debe tener mas practicas y una mejor enseñanza de este tema en particular.
- La guía de Java JFrame es una herramienta muy útil, la cual puede ser utilizada para empezar.

#### Conclusión:

- Aunque la creación de la interfaz y sus proceso, son complicados y es necesario practicar mucho, al igual que solicitar ayuda si es necesario, por otra parte

realizar este tipo de ejercicios es muy útil para la práctica de programación en java.