****

UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

***La Universidad Católica de Loja***

**PROGRAMACIÓN ORIENTADA A OBJETOS**

**Docente:**

Ing. Richar Stalin Guaya Delgado

**Estudiante:**

Roberto Narváez

**Tema:**

Proyecto punto de venta

**PERÍODO ACADEMICO EXTRARODINARIO FEBRERO 2020**

# **Link del repositorio.**

https://github.com/mrnarvaez3/proyectopoo.git

# **Descripción del problema.**

Elaborar un programa el cual presente un menú de opciones:

1. Registrar stock; donde se registrarán productos con sus respectivos atributos, precio, cantidad, nombre. Estos productos pueden ser de varios tipos: Comestibles, Utensilios, Oficina, Industriales.
2. Realizar venta; Se hace una búsqueda del producto ya en stock y seguidamente se procede a comprar una cantidad de dicho producto. Al realizar la venta del producto se debe ir presentando el subtotal, IVA, y total a pagar de la compra hecha.
3. Salir.

# **Análisis.**

## **Entrada:** nombreProducto, cantidadProducto, precioProducto, tipoProducto, cantidadCompra.

## **Procesos:** obtenerCalculos():

totalProd = cantidadCompra \* precioProducto

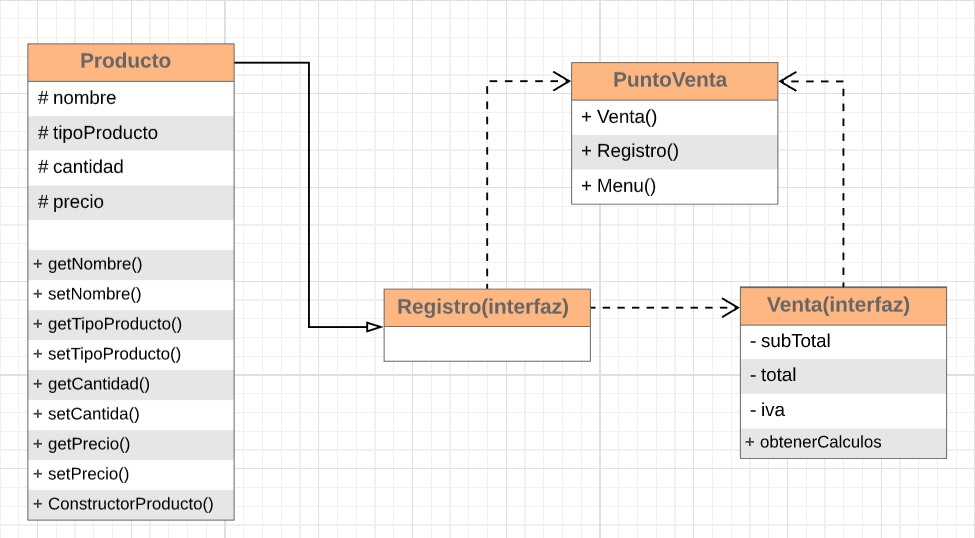
subTotal = subTotal + totalProd

IVA = subTotal \* 0.12

totalPagar = subTotal + IVA

## **Salida:** subTotal, IVA, totalPagar, nombreProducto, cantidadProducto, precioProducto, tipoProducto, cantidadCompra.

# **Diagrama de clases.**



# **Seudocódigo.**

Algoritmo punto de venta

Clase Producto

1. Declaracion de variables

# nombre: cadena

# tipoProducto: cadena

# cantidad: Real

# precio: Real

1. Método para obtenerNombre(): Cadena

retornar nombre

Fin método

1. Método establecerNombre(nombre: Cadena)

nombre = nombre

Fin metodo

1. Método para obtenerTipoProducto(): Cadena

retornar tipoProducto

Fin método

1. Método establecerTipoProducto (tipoProducto: Cadena)

tipoProducto = tipoProducto

Fin metodo

1. Método para obtenerCantidad(): Real

retornar cantidad

Fin método

1. Método establecerCantidad(cantidad : Real)

cantidad = cantidad

Fin metodo

1. Método para obtenerPrecio(): Real

retornar precio

Fin método

1. Método establecerPrecio(precio: Real)

precio = precio

Fin metodo

1. Método constructor Producto (nom: Cadena, cant: Real, pre: Real, tipo: Cadena)

nombre = nom

cantidad = cant

precio = pre

tipoProducto = tipo

Fin método constructor

Fin Clase

Clase PuntoVenta

1. Declaracion de variables tipo objeto

// Para llamar cada interfaz y así inicializarlas una única vez

Venta venta = new Venta();

Registro registro = new Registro();

Menu menu = new Menu();

Fin Clase

Clase interfaz Venta

1. Declaracion de variables

- subTotal: Real

- iva: Real

- total: Real

- totalProd: Real

2. Metodo obtenerCalculos()

totalProd = cantidadCompra \* precioProducto

subTotal = subTotal + totalProd

IVA = subTotal \* 0.12

totalPagar = subTotal + IVA

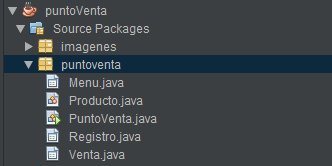
Fin metodo

Fin clase

# **Explicación del programa.**

* **Estructura del proyecto:**

El proyecto ya finalizado consta de 2 clases y 3 interfaces, la clase Producto es la cual obtiene y establece todos los atributos de mi objeto producto, la clase PuntoVenta se encarga de ejecutar las interfaces una única vez para que no se renueven las variables de cada lista. En las clases interfaces lo que tenemos son la visualización de menu principal, de registro de productos en stock, y de venta de productos.

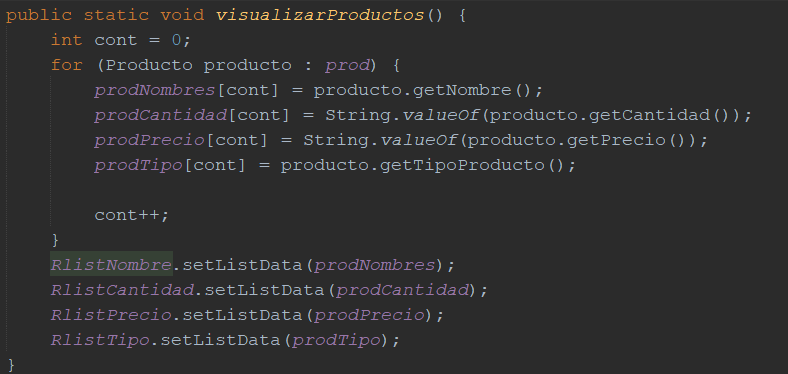


* **Interfaz Menu:**

En la interfaz menu lo que tenemos son dos botones; el primero que te redirige a la interfaz de registro de productos, el segundo a la interfaz de venta de productos y el tercero, el de salir que finaliza el programa.

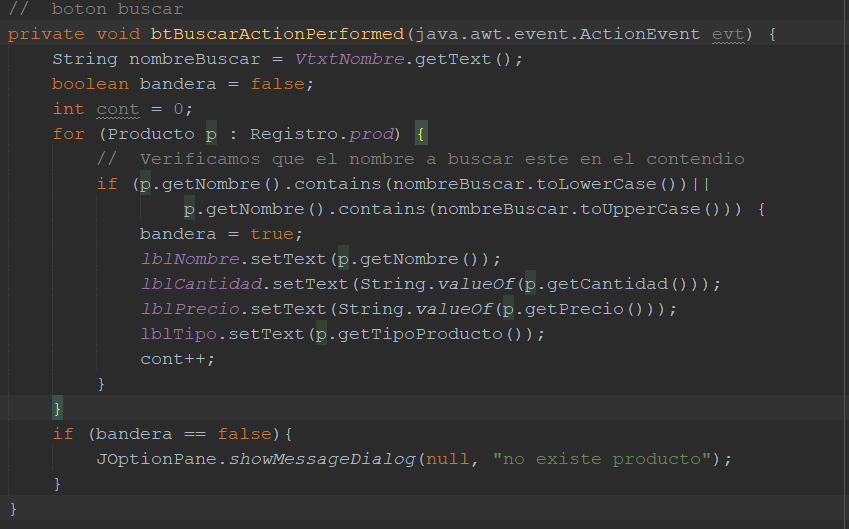
* **Interfaz Registro:**

En la interfaz Registro se encuentran una serie de elementos, entre ellos Labels, botones, textField, y listas. Se realizaron 4 Arrays de tipo Producto, uno para cada Jlist las cuales se fueron llenando mediante un for que recogía los datos de los TextField y se iban almacenando en un ArrayList el cual sirvió para seguidamente enviarlo a cada lista, cabe recalcar que cada Array y Jlist identifica cada atributo del producto, es decir nombre, cantidad, precio y tipo.

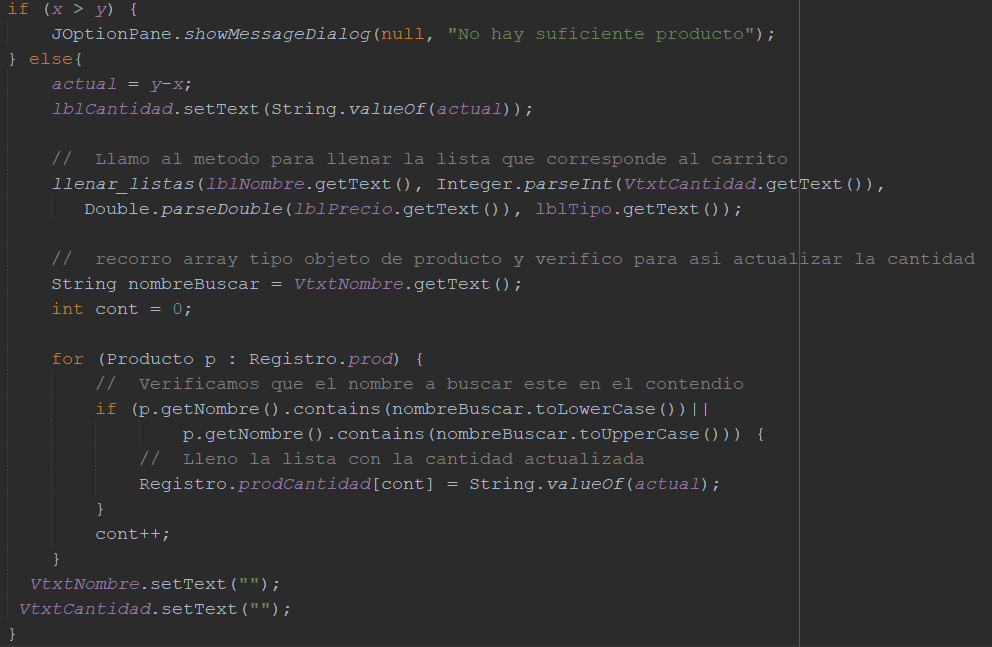


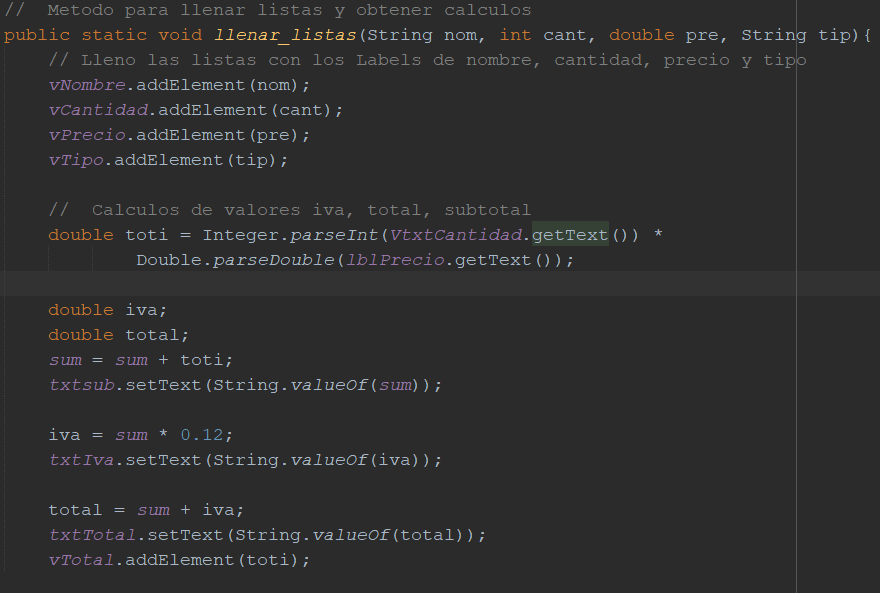
* **Interfaz Ventas:**

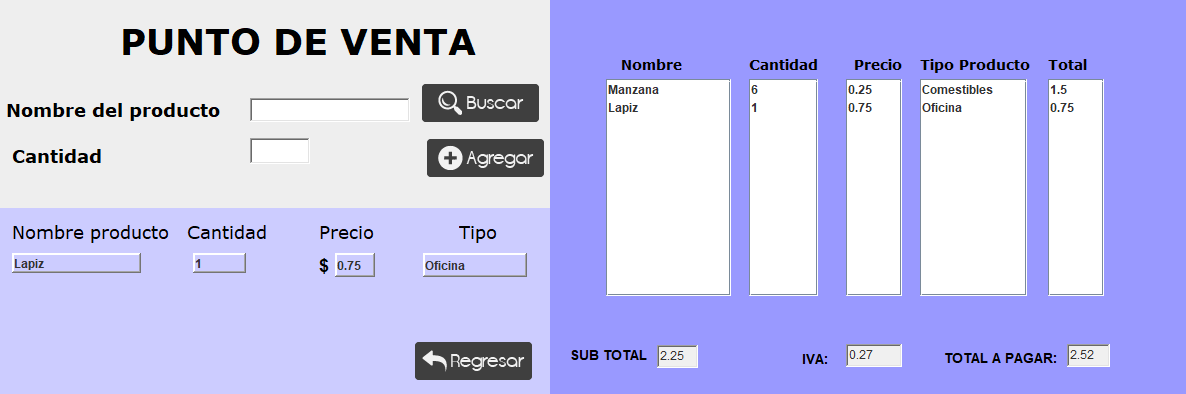
En la interfaz Ventas tenemos la mayoría de procesos, se obtiene las listas con el stock ya agregado, las cantidades del sub total, del iva y del total a pagar, y finalmente la actualización de la cantidad del producto al stock, ya que cuando se efectúa una compra el producto comprado debe disminuir a la cantidad que se vendió del producto.

El mecanismo o lógica que se implementó es la siguiente, Una vez se ingresa el nombre y le damos a buscar, en un ciclo for pregunto a cada elemento de mi listaNombre si contiene el texto por buscar. Una vez encuentra el elemento, lo sitúo en los Labels de la parte inferior.

Para realizar la compra del producto y agregarlo al “carrito” primeramente se verifica que la cantidad a comprar sea menor a la cantidad en stock del producto, seguidamente los datos encontrados por el botón buscar se llevaron a otras Jlist que cree para poder mostrar y obtener los cálculos y finalmente lleno la listaCantidad del stock con la cantidad ya actualizada.



A continuación, el código de cómo se llenaron las listas del producto a comprar y como obtuve los cálculos del sub total, iva y total a pagar:

Y el resultado final de la interfaz Venta es el siguiente.