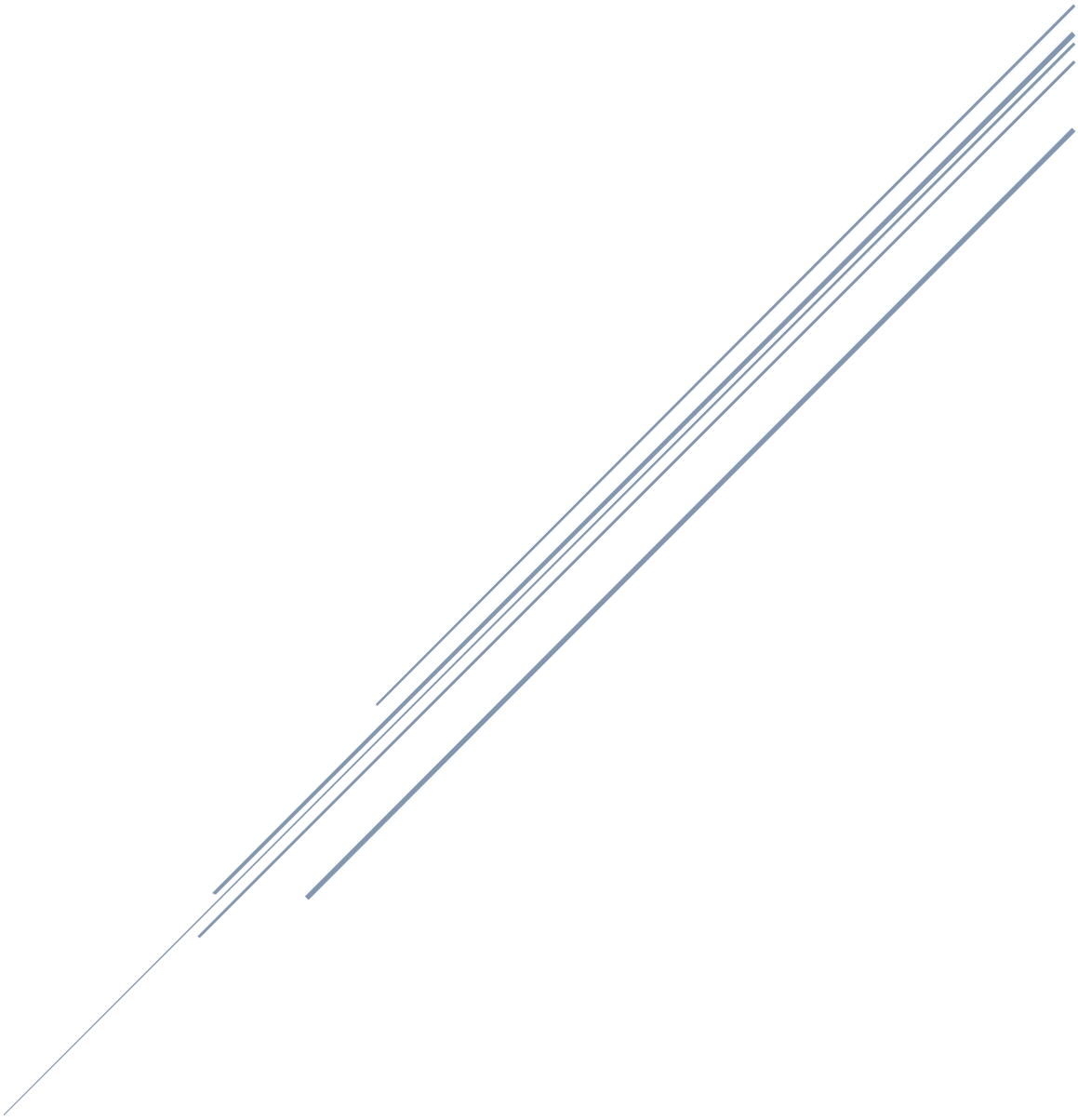
MANUAL TÉCNICO

BLUE MALL - POS



Carlos Javier Cox Bautista
IPC1

Introducción

El presente documento describe los aspectos técnicos informáticos del sistema de información. El documento familiariza al personal técnico especializado encargado de las actividades de administración, Vendedores, Supervisión de Gastos.

El sistema está compuesto de forma que permita que la información contenida pueda ser agregada, editada y/o modificada por el personal encargado del departamento o administrador. Para alcanzar estos propósitos se ha hecho uso de JAVA que es un lenguaje que se adecua a las nuevas necesidades de las aplicaciones actuales, a través de esta tecnología la administración y seguridad de la información se dará de forma centralizada y segura de datos, al mismo tiempo facilita la actualización eficiente de dicha información.

En cuanto a este manual se ha considerado incluir todos los aspectos técnicos necesarios para el manejo y control de la aplicación para el personal administrativo y el personal de Vendedores.

Dirigido Para:

Este manual está orientado a los Vendedores y personal encargado de Administrar la cantidad de Productos, Clientes, Vendedores y sucursales de la compañía. Solamente dichas personas están autorizadas a realizar modificaciones en el sistema.

Objetivos

General

El objetivo principal de implementar un sistema POS en una tienda de es el de reducir los problemas que se tienen a la hora de gestionar toda la información de todas la variables que la Tienda BLUE MALL maneja, y permitiéndonos tener un mejor control de su inventario y registran todas las actividades de una tienda.

Específicos

- Manejar La información de las sucursales, vendedores, clientes y productos de un amanera visual y fácil de entender.
- Agilizar la forma de guardar las ventas realizadas por cada vendedor.
- Crear Facturas de forma Rápida y Sencilla

Especificación técnica

Hardware

Procesador: de 1 gigahercio (GHz), o procesador o SoC más rápido

RAM: 1 gigabyte (GB) para 32 bits o 2 GB para 64 bits

Espacio en disco duro:16 GB para el sistema operativo de 32 bits o 20 GB para el sistema operativo de 64 bits

Tarjeta gráfica: DirectX 9 o posterior con controlador WDDM 1.0

Pantalla: 800 x 600 o superior

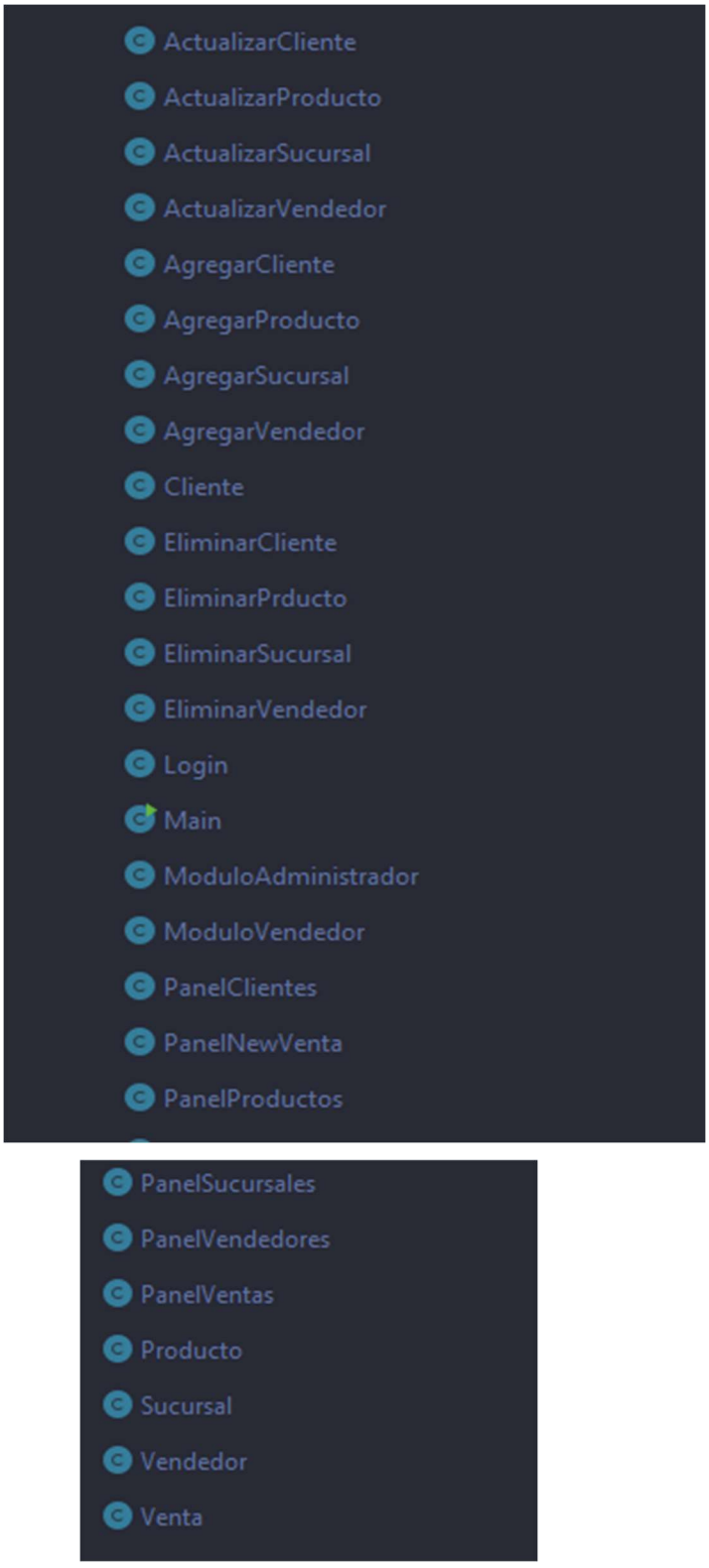
Software

Último sistema operativo: Asegúrate de que estás ejecutando la versión más reciente: Windows 8.1 o Superior.

Tener Instalado Java 8.0

Lógica Del Sistema

Clases Utilizadas



Se Hizo uso de un total de 29 Clases utilizadas De las cuales entre ellas se puede dividir en:

- Agregar
- Actualizar
- Eliminar
- Módulos
- Paneles
- Objetos

Para el apartado **Agregar**, **Eliminar y Actualizar** dentro de estas clases heredan de la Clase JPFrame ya que contiene características de un Frame dentro de java, dentro de ellas contiene lo siguiente.

public Agregarxxxx (: Este constructor es el que crea todos los componentes que el Jframe va a contener, como por ejemplo los Jlabel y JtextField.

public void agregarXxxxx(): Este método es el encargado deir agregando la información introducida por el usuario al Vector correspondiente.

public void EliminarXXX(): Este método es el encargado de ir eliminando la información introducida por el usuario al Vector correspondiente.

public void ActXXXX(): Este método es el encargado de ir modificando la la información introducida por el usuario al Vector correspondiente con el código verificado el código de identificación del producto, cliente, sucursal…etccc.

```
public class ActualizarProducto extends JFrame implements ActionListener {
    JLabel lbltitulo,lblCodigo,lblNombre,lblDescripcion,lblCantidad,lblPrecio,lblInfo;
    JTextField txtCodigo,txtNombre,txtDescripcion,txtCantidad,txtPrecio;
    JButton btnAct, btnCancelar;
    int posProducto;

    public ActualizarProducto() {...}


    public void Act(int pos) throws IOException {...}


    @Override
    public void actionPerformed(ActionEvent actionEvent) {...}
}
```

```
    JLabel lbltitulo,lblCodigo,lblNombre,lblDescripcion,lblCantidad,lblPrecio;
    JTextField txtCodigo,txtNombre,txtDescripcion,txtCantidad,txtPrecio;
    JButton btnAgregar, btnCancelar;

    public AgregarProducto(){...}

    public  void agregarProducto() throws IOException {...}
```

Para el apartado de módulos están **Modulo Aministrador y Modulo vendedor**, los cuales entre ellos manejan toda la información y los objetos creados anteriormente agregando uno por uno conforme el usuario interactúa con cada uno de ellos.

```
public class ModuloAdministrador extends JFrame implements ActionListener {

    JTabbedPane paneles = new JTabbedPane();
    public ModuloAdministrador(){
        Main.panelSucursales = new PanelSucursales();
        Main.panelProductos = new PanelProductos();
        Main.panelClientes = new PanelClientes();
        Main.panelVendedores = new PanelVendedores();
        paneles.addTab( title: "Sucursales", Main.panelSucursales);
        paneles.addTab( title: "Productos", Main.panelProductos);
        paneles.addTab( title: "Clientes", Main.panelClientes);
        paneles.addTab( title: "Vendedores", Main.panelVendedores);
        this.setTitle("Módulo Administrador");
        this.setBounds( x: 150, y: 20, width: 1000, height: 600);
        this.getContentPane().add(paneles);
        this.setResizable(false);
        this.setVisible(true);
        this.setLayout(null);
        this.addWindowListener((WindowAdapter) windowClosing(windowEvent) → {
                JOptionPane.showMessageDialog(Main.ventanaLogin, message: "Se ha cerrado sesión.");
                Main.ventanaLogin.setVisible(true);
                Main.ventanaLogin.txtUsuario.requestFocus();
        });
```

```
public class ModuloVendedor extends JFrame implements ActionListener {

    JTabbedPane paneles = new JTabbedPane();
    Vendedor vendedor;
    int codigoVendedor;
public ModuloVendedor(int codigoVendedor){
    vendedor = Main.vendedores[Main.buscarCodigoVendedor(codigoVendedor)];
    this.codigoVendedor = codigoVendedor;
    Main.panelNewVenta = new PanelNewVenta(vendedor,this.codigoVendedor);
    Main.panelVentas = new PanelVentas(vendedor,this.codigoVendedor);

    paneles.addTab( title: "Nueva venta", Main.panelNewVenta);
  paneles.addTab( title: "Ventas", Main.panelVentas);

    this.setTitle("Módulo Vendedor - " + Main.vendedores[codigoVendedor] );
    this.setBounds( x: 150, y: 20, width: 1000, height: 600);
    this.getContentPane().add(paneles);
    this.setResizable(false);
    this.setVisible(true);
    this.setLayout(null);
```

Para el apartado de Objetos están Cliente,Vendedor,Venta,Sucursal y producto los cuales entre ellos manejan toda la información y características de cada una de las anteriores mencionadas para hacer referencia y obtener los datos de una manera mas sencilla.

Cada uno de Estos contiene su contructor con las característica y sus Getters y Setters para modificar dichas características

```
public class Sucursal implements Serializable{

    private int codigo;
    private String nombre;
    private String direccion;
    private String correo;
    private String telefono;
    public Sucursal(int codigo, String nombre, String direccion, String correo, String telefono) {
        this.codigo = codigo;
        this.nombre = nombre;
        this.direccion = direccion;
        this.correo = correo;
        this.telefono = telefono;
    }

    public int getCodigo() { return codigo; }

    public void setCodigo(int codigo) { this.codigo = codigo; }

    public String getNombre() { return nombre; }

    public void setNombre(String nombre) { this.nombre = nombre; }
```