

## MANUAL TECNICO

**Nombre del Programa:** Becksport, Programa de inventario

**Version:**1.0

**Ultima Actualización:** 09 de noviembre 2024

**Autor:** Wilson Ronaldo Cúmez Otzoy

**Cargo:** Estudiante de Ingeniería en sistemas

**Nota:** La presentación final del programa podría variar respecto de las imágenes proporcionadas en el manual, esto debido a las actualizaciones que se estuvieron realizando durante la creación del programa.

### **Propósito del manual:**

Este busca poder presentar al usuario el desarrollo del programa durante su tiempo de creación a través de imágenes del código y diseño utilizado para completar cada proceso. Con base a este manual se espera que los desarrolladores pueden tener un buen entendimiento de la estructura del programa.

### **Descripción del programa:**

Este sistema permite registrar información detallada de cada producto, como código, nombre, cantidad y precio unitario, proporcionando a los usuarios una vista clara y precisa del estado de inventario. Al automatizar y centralizar estos datos, el sistema reduce errores manuales, mejora la toma de decisiones mediante reportes actualizados y ayuda a mantener un flujo de stock adecuado, evitando faltantes y excesos.

### **Tecnologías y herramientas usadas:**

- NetBeans IDE 8.2
- Lenguaje de programación Java
- iText 5.5.12
- FlatIcon
- Mysql workbench

## Creación de la Base de datos de Log in y producto:

```
• create database DBBecksport;

• use DBBecksport;
• create table producto(
  codigoProducto varchar(10) primary key,
  nombreProducto varchar(30) not null,
  precioUnitario decimal(10,2) not null,
  cantidadProducto int not null
);

/*Insertando datos*/
• insert into producto(codigoProducto,nombreProducto,precioUnitario,cantidadProducto)
  values('4','Media futbol-negro','300','100');
```

```
create database DBbecksportLogin;

use DBbecksportLogin;
create table usuario(
id int(10) primary key,
email varchar(30) not null,
clave varchar(30) not null
);
```

## Conexión a la base de datos:

```

public class LConexion {

    String user ="root";
    String password ="";
    String bd ="DBbecksportLogin";
    String ip ="localhost";
    String puerto ="3306";

    String cadena = "jdbc:mysql://" +ip+": "+puerto+"/" +bd;

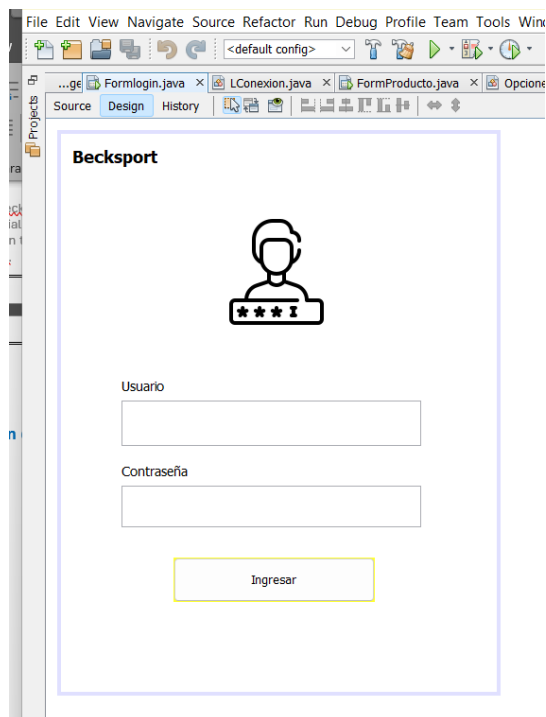
    public Connection estableceConexion() {

        Connection con= null;

        try{
            con=DriverManager.getConnection(cadena,user,password);
        }catch(SQLException e){
        }
        return con;
    }
}

```

## Creación de la ventana FormLogin:



## Parte del código fuente:

```
Connection cn=objetoConexion.estableceConexion();

if (conexion != null) {
    try {

        String query = "SELECT * FROM usuario WHERE email = ? AND clave = ?";
        PreparedStatement ps = conexion.prepareStatement(query);
        ps.setString(1, usuario);
        ps.setString(2, pass);

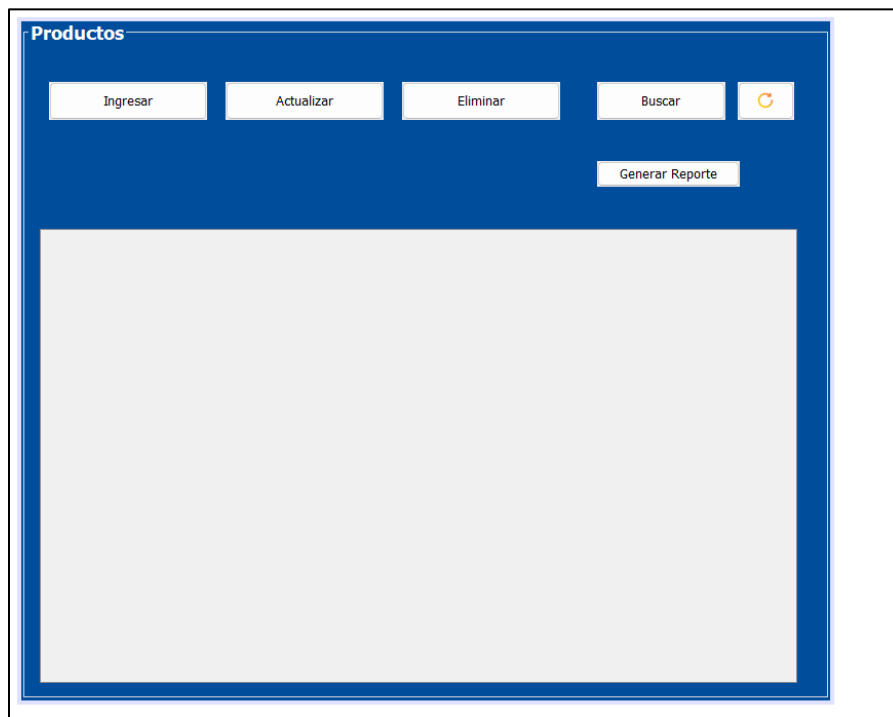
        ResultSet rs = ps.executeQuery();

        if (rs.next()) {
            JOptionPane.showMessageDialog(this, ";Bienvenido " + usuario + "!");

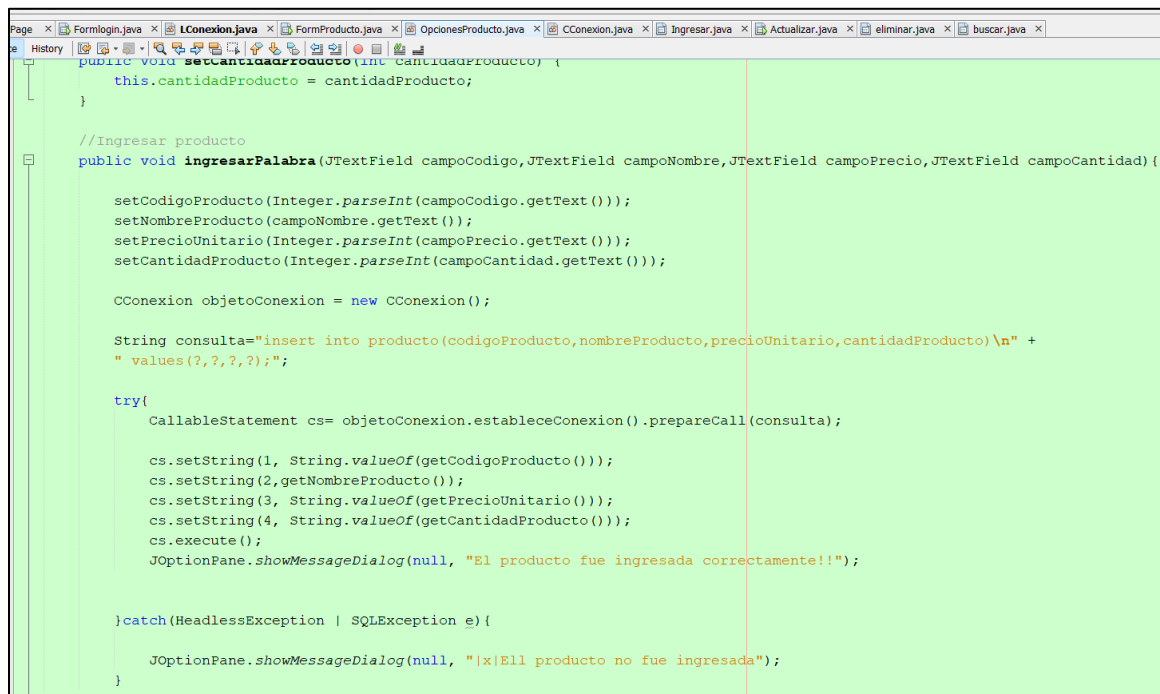
            java.awt.EventQueue.invokeLater(() -> {
                new FormProducto().setVisible(true);
            });

            dispose();//cerrando la ventana de login
        } else {
            JOptionPane.showMessageDialog(this, "Usuario o contraseña incorrectos");
        }
    }
}
```

## Creación de la Ventana principal FormProducto:



Parte del código fuente proveniente tanto de FormProducto.java y OpcionesProducto.java:



```
Page x Formlogin.java x LConexion.java x FormProducto.java x OpcionesProducto.java x CConexion.java x Ingresar.java x Actualizar.java x eliminar.java x buscar.java x
History
public void setCantidadProducto(int cantidadProducto) {
    this.cantidadProducto = cantidadProducto;
}

//Ingresar producto
public void ingresarPalabra(JTextField campoCodigo, JTextField campoNombre, JTextField campoPrecio, JTextField campoCantidad) {

    setCodigoProducto(Integer.parseInt(campoCodigo.getText()));
    setNombreProducto(campoNombre.getText());
    setPrecioUnitario(Integer.parseInt(campoPrecio.getText()));
    setCantidadProducto(Integer.parseInt(campoCantidad.getText()));

    CConexion objetoConexion = new CConexion();

    String consulta="insert into producto(codigoProducto,nombreProducto,precioUnitario,cantidadProducto)\n" +
        " values(?, ?, ?, ?)";

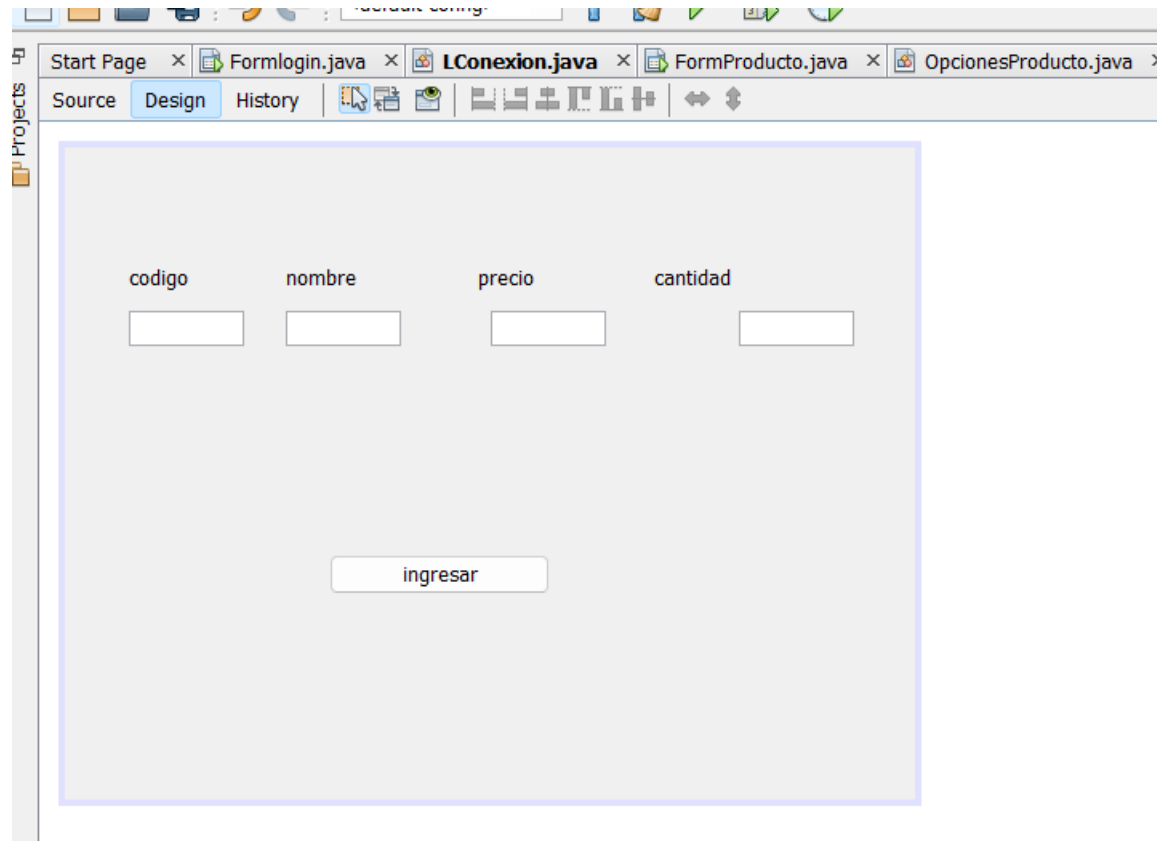
    try{
        CallableStatement cs= objetoConexion.estableceConexion().prepareCall(consulta);

        cs.setString(1, String.valueOf(getCodigoProducto()));
        cs.setString(2, getNombreProducto());
        cs.setString(3, String.valueOf(getPrecioUnitario()));
        cs.setString(4, String.valueOf(getCantidadProducto()));
        cs.execute();
        JOptionPane.showMessageDialog(null, "El producto fue ingresada correctamente!!");

    }catch (HeadlessException | SQLException e){

        JOptionPane.showMessageDialog(null, "|x|El producto no fue ingresada");
    }
}
```

**Creación de las ventanas para Ingresar, Actualizar, Eliminar, Buscar:**



Parte del código para cada ventana:

```
public void buscarPorCodigo(JTextField campoCodigo, JTextField campoNombre, JTextField campoPrecio, JTextField campoCantidad) {  
  
    String consulta="select codigoProducto,nombreProducto,precioUnitario,cantidadProducto from producto where producto.codigoProducto=?";  
  
    CConexion objetoConexion = new CConexion();  
    try{  
        CallableStatement cs = objetoConexion.estableceConexion().prepareCall(consulta);  
        cs.setString(1,campoCodigo.getText());  
        cs.execute();  
  
        ResultSet rs = cs.executeQuery();  
  
        if(rs.next()){  
  
            JOptionPane.showMessageDialog(null,"Registro encontrado");  
            campoCodigo.setText(rs.getString("codigoProducto"));  
            campoNombre.setText(rs.getString("nombreProducto"));  
            campoPrecio.setText(rs.getString("precioUnitario"));  
            campoCantidad.setText(rs.getString("cantidadProducto"));  
  
        }else{  
  

```