

Sistema de facturación e inventario de la tienda de abarrotes “Paskeline”

Fabiola Bucheli¹; Edgar Caicedo²

¹ Universidad Técnica Particular de Loja, Quito-Ecuador, fjgl_02@hotmail.com, ² Universidad Tecnológica Israel, Quito-Ecuador, ecaicedo@uisrael.edu.ec, msalazar@uisrael.edu.ec

Resumen: El sistema de facturación e inventario de la tienda de abarrotes Paskeline es un sistema que consta de diversas funciones, entre ellas está el poder registrar un empleado para que pueda hacer uso del sistema, podemos también registrar los productos para tener un control del stock de la tienda, este stock también podrá ser consultado por cualquier trabajador para poder vender el producto, también se podrá registrar a los diversos proveedores de productos que abastece a la tienda, nuestro sistema tiene una interfaz fácil de manejar para el usuario y está desarrollada en lenguaje de programación java y utiliza el ide de netbeans como compilador; es compatible con Windows 7 y posteriores; utiliza como base de datos MySQL

Palabras clave: Interfaz, stock, ide, java, netbeans, programación, compilador, Windows, MySQL.

Radius Server Application on Linux Mint Distribution

Abstract: The billing and inventory system of the Paskeline grocery store is a system that consists of various functions, among them is the ability to register an employee so that they can use the system, we can also register the products to have control of the stock of the store. store, this stock can also be consulted by any worker to be able to sell the product, it will also be possible to register the various suppliers of products that supply the store, our system has an easy-to-use interface for the user and is developed in language of java programming and uses the netbeans ide as a compiler; is compatible with Windows 7 and later; uses MySQL as database.

Keywords: Interface, stock, ide, java, netbeans, programming, compiler, Windows, MySQL.

1. INTRODUCCIÓN

En la actualidad la tienda de abarrotes “Paskeline” no consta de un sistema de facturación ni registro de mercadería, por lo que se propone crear una aplicación que logre realizar facturación de los productos en venta y el registro de los mismos a la hora de abastecer la tienda.

Se requiere construir una aplicación que permita realizar todo el proceso de facturación e inventario en un almacén, el sistema debe permitir realizar los siguientes procesos de registro:

- Registro de clientes
- Registro de artículos
- Registro de proveedores

Además, se deben listar los respectivos procesos construidos anteriormente.

El sistema debe poder actualizar el stock o las existencias de los artículos, a lo que llamaremos inventario, esto con el fin de mantener el proceso de

facturación acorde a los artículos existentes en bodega. El proceso más importante que se debe realizar es el de facturación, que implica generar una factura y adicionar artículos de acuerdo a las cantidades requeridas, además calcular los valores totales y de impuestos automáticamente.

2. METODOLOGÍA

La metodología Lean considerada tanto una metodología de trabajo como una filosofía centrada en maximizar el valor del cliente y minimizar el desperdicio. Esto traducido a los procesos de fabricación “pull” radica en producir solo lo necesario y en el momento adecuado. (Incubicon, 2019)

2.1. Planificación

Del levantamiento del proyecto se requiere una aplicación que permita el registro de clientes, artículos y proveedores además de esto se va a poder actualizar el stock, esto con el fin de mantener el proceso de facturación acorde a los artículos existentes, el proceso más importante que se debe realizar es el de facturación, que implica generar una factura y adicionar artículos de acuerdo a las cantidades requeridas, además calcular los valores totales y de impuestos automáticamente.

2.2. Diseño

En este primer diseño, se realiza una interfaz práctica para un usuario, basándose en una buena práctica de interfaz de usuario con parámetros básicos de configuración.

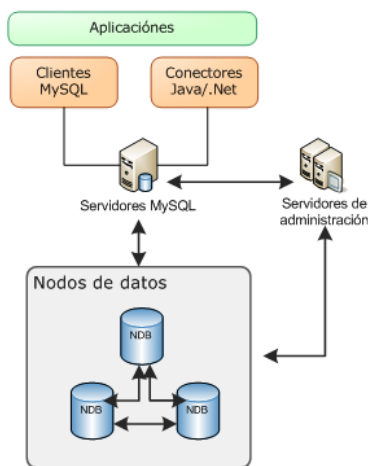


Ilustración 1. MySQL Cluster.
 Fuente: (Chrome, 2020)

MySQL es el sistema de gestión de bases de datos relacional más extendido en la actualidad al estar basada en código abierto.

Este protocolo se utiliza globalmente para la transmisión de contenido en diferentes formatos hipertexto, videos, imágenes, audio, etc.

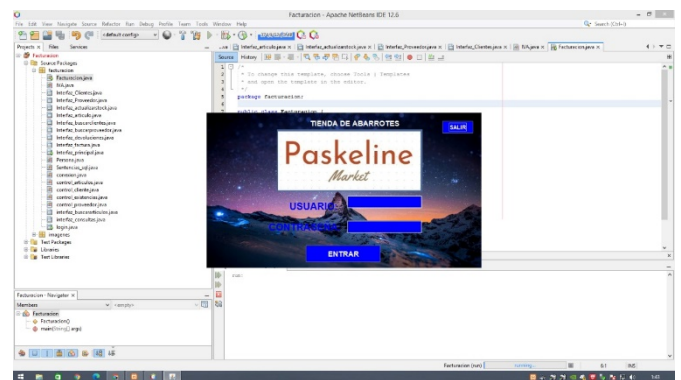
Las solicitudes constan de los siguientes elementos:

- Los métodos java GET, SET que sirve para el envío y recepción de datos. Por lo general, un cliente desea obtener un recurso (usando GET).

- Solicitudes para la comunicación de la base de datos de tipo string.
- Conector SQL para la conexión entre el compilador con la base de datos.

2.3. Desarrollo

El diseño de la interfaz contiene un cuadro de texto para introducir el usuario y la contraseña del empleado, como parámetros de configuración la inclusión de imágenes y documentos y el tiempo de espera del sitio web para su carga. Como resultados incluimos el login de la aplicación.



```
public class login extends javax.swing.JFrame {
    /**
     * Creates new form login
     */
    public login() {
        initComponents();
        this.setLocationRelativeTo(null);
    }
    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code
    private void btnentrarActionPerformed(java.awt.event.ActionEvent evt) {
        String usu=txtusuario1.getText();
        String pas=new String(jpascontraseña.getPassword());
        if(usu.equals("admin") && pas.equals("tiendak"))
        {
            this.setVisible(false);
            Interfaz_principal ingreso = new Interfaz_principal ();
            ingreso.setVisible(true);
            ingreso.pack();
        }
        else
        {
            JOptionPane.showMessageDialog(null, "Los Datos son Incorrectos vuelva a Intentarlo");
            txtusuario1.requestFocus();
        }
    }
}
```

Se está realizando la validación de la contraseña mediante un método jpascontraseña.

```
String usu=txtusuario1.getText();

String pas=new
String(jpascontraseña.getPassword());

if(usu.equals("admin") &&
pas.equals("tiendak"))
```

Proceso de facturación e inventario

```
package facturacion;

import javax.swing.JOptionPane;

public class login extends javax.swing.JFrame {

    /**
     * Creates new form login
     */
    public login() {
        initComponents();
        this.setLocationRelativeTo(null);
    }

    /**
     * This method is called from within the
     constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content
     of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed"
    desc="Generated Code">
    private void initComponents() {

        jLabel3 = new javax.swing.JLabel();
        jLabel1 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        txtusuario1 = new javax.swing.JTextField();
        btnentrar = new javax.swing.JButton();
        jpascontraseña = new
        javax.swing.JPasswordField();
        btnsalir = new javax.swing.JButton();
        jLabel5 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLayeredPane1 = new
        javax.swing.JLayeredPane();

        setDefaultCloseOperation(javax.swing.WindowConstants.
        EXIT_ON_CLOSE);
        setUndecorated(true);
        getContentPane().setLayout(new
        org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabel3.setIcon(new
        javax.swing.ImageIcon(getClass().getResource("/imagenes/tienda.jpeg"))); // NOI18N
        getContentPane().add(jLabel3, new
        org.netbeans.lib.awtextra.AbsoluteConstraints(170,
        40, 360, 150));

        jLabel1.setFont(new
        java.awt.Font("Transformers Movie", 1, 24)); //
        NOI18N
        jLabel1.setForeground(new java.awt.Color(0,
        255));
        jLabel1.setText("CONTRASEÑA:");
        getContentPane().add(jLabel1, new
        org.netbeans.lib.awtextra.AbsoluteConstraints(150,
        260, 180, 30));

        jLabel4.setFont(new
        java.awt.Font("Transformers Movie", 1, 24)); //
        NOI18N
        jLabel4.setForeground(new java.awt.Color(0,
        255));
        jLabel4.setText("USUARIO:");
```

```
        getContentPane().add(jLabel4, new
        org.netbeans.lib.awtextra.AbsoluteConstraints(200,
        210, 120, 30));

        txtusuario1.setBackground(new
        java.awt.Color(0, 0, 255));
        txtusuario1.setFont(new
        java.awt.Font("Transformers Movie", 1, 14)); //
        NOI18N
        txtusuario1.setForeground(new
        java.awt.Color(255, 255, 255));
        txtusuario1.addActionListener(new
        java.awt.event.ActionListener() {
            public void
            actionPerformed(java.awt.event.ActionEvent evt) {
                txtusuario1ActionPerformed(evt);
            }
        });
        getContentPane().add(txtusuario1, new
        org.netbeans.lib.awtextra.AbsoluteConstraints(340,
        200, 180, 30));

        btnentrar.setBackground(new java.awt.Color(0,
        255));
        btnentrar.setFont(new
        java.awt.Font("Transformers Movie", 1, 18)); //
        NOI18N
        btnentrar.setForeground(new
        java.awt.Color(255, 255, 255));
        btnentrar.setText("ENTRAR");
        btnentrar.addActionListener(new
        java.awt.event.ActionListener() {
            public void
            actionPerformed(java.awt.event.ActionEvent evt) {
                btnentrarActionPerformed(evt);
            }
        });
        getContentPane().add(btnentrar, new
        org.netbeans.lib.awtextra.AbsoluteConstraints(240,
        320, 180, 40));

        jpascontraseña.setBackground(new
        java.awt.Color(0, 0, 255));
        jpascontraseña.setFont(new
        java.awt.Font("Transformers", 1, 14)); // NOI18N
        jpascontraseña.setForeground(new
        java.awt.Color(255, 255, 255));
        getContentPane().add(jpascontraseña, new
        org.netbeans.lib.awtextra.AbsoluteConstraints(340,
        260, 180, 30));

        btnsalir.setBackground(new java.awt.Color(0,
        255));
        btnsalir.setFont(new
        java.awt.Font("Transformers Movie", 1, 14)); //
        NOI18N
        btnsalir.setForeground(new
        java.awt.Color(255, 255, 255));
        btnsalir.setText("SALIR");
        btnsalir.addActionListener(new
        java.awt.event.ActionListener() {
            public void
            actionPerformed(java.awt.event.ActionEvent evt) {
                btnsalirActionPerformed(evt);
            }
        });
        getContentPane().add(btnsalir, new
        org.netbeans.lib.awtextra.AbsoluteConstraints(570,
        20, -1, -1));

        jLabel5.setFont(new
        java.awt.Font("Transformers Movie", 1, 18)); //
        NOI18N
```

```

        jLabel5.setForeground(new java.awt.Color(255,
255, 255));
        jLabel5.setText("TIENDA DE ABARROTES");
        getContentPane().add(jLabel5, new
org.netbeans.lib.awtextra.AbsoluteConstraints(250,
10, 240, 30));

        jLabel2.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagen
es/wallpaper-664x374.jpg"))); // NOI18N
        getContentPane().add(jLabel2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -
1, -1));
        getContentPane().add(jLayeredPane1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(340,
70, -1, -1));

        pack();
    } // </editor-fold>

    private void
btnentrarActionPerformed(java.awt.event.ActionEvent
evt) {
        String usu=txtusuario1.getText();
        String pas=new
String(jpascontraseña.getPassword());
        if(usu.equals("admin") && pas.equals("tiendak"))
        {
            this.setVisible(false);
            Interfaz_principal ingreso = new
Interfaz_principal ();
            ingreso.setVisible(true);
            ingreso.pack();

        }
        else
        {
            JOptionPane.showMessageDialog(null, "Los
Datos son Incorrectos vuelva a Intentarlo");
            txtusuario1.requestFocus();
        }
    }

    private void
btnsalirActionPerformed(java.awt.event.ActionEvent
evt) {
        System.exit(0);
    }

    private void
txtusuario1ActionPerformed(java.awt.event.ActionEvent
evt) {
        // TODO add your handling code here:
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed"
desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not
available, stay with the default look and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lo
okandfeel/plaf.html
         */
        try {
            for
(javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {

```

```

                if ("Nimbus".equals(info.getName()))
                {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            } catch (ClassNotFoundException ex) {
                java.util.logging.Logger.getLogger(login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
            } catch (InstantiationException ex) {
                java.util.logging.Logger.getLogger(login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
            } catch (IllegalAccessException ex) {
                java.util.logging.Logger.getLogger(login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
            } catch (javax.swing.UnsupportedLookAndFeelException ex) {
                java.util.logging.Logger.getLogger(login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
            }
        } //</editor-fold>
        //</editor-fold>

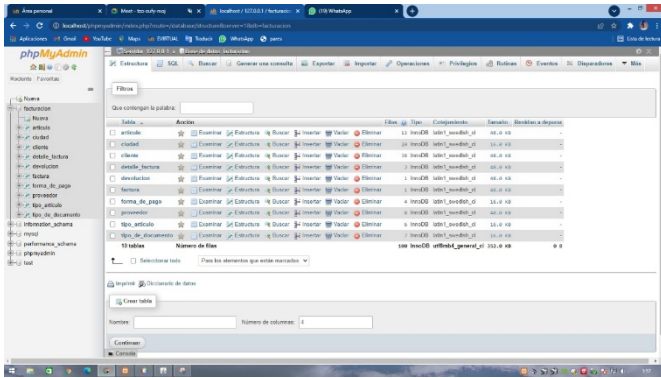
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new
Runnable() {
            public void run() {
                new login().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton btnentrar;
    private javax.swing.JButton btnsalir;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLayeredPane jLayeredPane1;
    private javax.swing.JPasswordField
jpascontraseña;
    private javax.swing.JTextField txtusuario1;
    // End of variables declaration
}

```

2.4.Pruebas

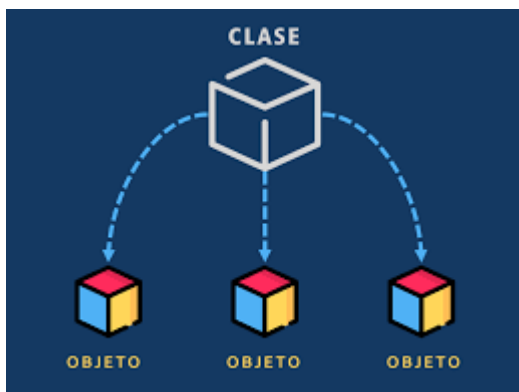
El ingreso y validación de los datos se ve reflejaba en la base de datos.



3. RESULTADOS Y DISCUSIÓN

3.1. Identificación de contenido.

La programación orientada a objetos (Object Oriented Programming, OOP) es un modelo de programación informática que organiza el diseño de software en torno a datos u objetos, en lugar de funciones y lógica. Un objeto se puede definir como un campo de datos que tiene atributos y comportamiento únicos.



La programación orientada a objetos se centra en los objetos que los desarrolladores quieren manipular en lugar de enfocarse en la lógica necesaria para manipularlos. Este enfoque de programación es adecuado para programas que son grandes, complejos y se actualizan o mantienen activamente.

3.2. Principios de OOP

La programación orientada a objetos se basa en los siguientes principios:

Encapsulación. La implementación y el estado de cada objeto se mantienen de forma privada dentro

de un límite definido o clase. Otros objetos no tienen acceso a esta clase o la autoridad para realizar cambios, pero pueden llamar a una lista de funciones o métodos públicos. Esta característica de ocultación de datos proporciona una mayor seguridad al programa y evita la corrupción de datos no intencionada.

Abstracción. Los objetos solo revelan mecanismos internos que son relevantes para el uso de otros objetos, ocultando cualquier código de implementación innecesario. Este concepto ayuda a los desarrolladores a realizar cambios y adiciones más fácilmente a lo largo del tiempo.

Herencia. Se pueden asignar relaciones y subclases entre objetos, lo que permite a los desarrolladores reutilizar una lógica común sin dejar de mantener una jerarquía única. Esta propiedad de OOP obliga a un análisis de datos más completo, reduce el tiempo de desarrollo y asegura un mayor nivel de precisión.

Polimorfismo. Los objetos pueden adoptar más de una forma según el contexto. El programa determinará qué significado o uso es necesario para cada ejecución de ese objeto, reduciendo la necesidad de duplicar código. (TechTarget, 2022)

3.3. Resultados.

Los resultados obtenidos fueron favorables ya que pudimos logearnos en el sistema con un nombre de usuario y una contraseña, además pudimos hacer el ingreso de cliente, personal y podemos tener acceso al stock de la tienda para saber qué productos están disponibles y que productos serán solicitados a un proveedor el cual también estará registrado en el sistema y toda esta información será guardada en una base de datos.

4. CONCLUSIONES

- La programación orientada a objetos es un método muy efectivo para la programación de aplicaciones de escritorio porque nos permite organizar el diseño de software en

torno a datos u objetos, en lugar de funciones y lógica.

- La programación orientada a objetos se asemeja al mundo real y fomenta la reutilización y ampliación del código
- Podemos concluir que la programación orientada a objetos (POO) es una forma especial de programar, más cercana a como expresaríamos las cosas en la vida real que otros tipos de programación. Tenemos que aprender a pensar las cosas de una manera distinta, para escribir nuestros programas.

REFERENCIAS

- Centro Europeo de Empresas e Innovación de Ciudad Real (2007). Manual de innovación: Guía Práctica de Gestión de la I+D+I para Pymes. Lince Artes Gráficas. España
- Champagne (2018). Blockchain España: El libro de Satoshi. Consultado el 03 de febrero <https://libroblockchain.com/wp-content/uploads/2018/07/Libro-deSatoshi-Blockchain-Espana-v1-junio-2018.pdf>
- Chicaiza, K. (2020). Determinantes de la innovación incremental y radical de las empresas ecuatorianas, período 2012-2014 (Bachelor's thesis, Quito, 2020.).
- Christensen, C., Raynor, M., y McDonald, R. (2015). What is Disruptive Innovation? Harvard Business Review, 93(12), 44-53. <https://doi.org/10.1353/abr.2012.0147>
- Coccia, M. (2017). Sources of technological innovation: Radical and incremental innovation problem-driven to support competitive advantage of firms. Technology Analysis & Strategic Management, 29(9), 1048-1061.
- Cruz, O. (2017). Innovación disruptiva: aportes conceptuales para organizaciones en Latinoamérica.
- Universidad Nacional de Colombia Ellis, S., & Brown, M. (2018). El Método Hacking Growth.
- Conecta Esic Business & Marketing School (2018). Que es la industria 4.0: transformación digital industrial. Consultado el 02 de Febrero <https://www.esic.edu/rethink/tecnologia/la-industria-4-0-transformacion-digitalindustrial>
- Fernández, F. J. L., & Rodríguez, J. C. F. (2018). La metodología Lean Startup: desarrollo y aplicación para el emprendimiento. Revista Escuela de Administración de Negocios, (84).
- Flash Co. (14 julio, 2021). ISO 56002: el sistema de gestión de la innovación definitivo. Consultado en <https://theflashco.com/iso-56002-el-sistema-de-gestion-de-lainnovacion-definitivo/>
- Gardner, H. (2016). Estructuras de la mente: la teoría de las inteligencias múltiples. Fondo de cultura económica.