

# MANUAL TÉCNICO

- 
- - *Super y Carnicería*
  - *“La Mexicana”*
  - 
  - *Villalobos Lázaro*
  - *Alexander*
  -
- 



Tabla de contenido

ESTE MANUAL .....	3
INSTALACIÓN DE MySQL Workbench Y NETBEANS8.2 .....	4
CARPETA la.mexicana.....	17
class LaMexicana.....	17
Panel 1 .....	17
MENU2.....	19
MENU .....	20
LLENADO.....	21
MENUAGG.....	26
AGREGARRES.....	27
MENUELIM .....	31
ELIMINARRES .....	32
BASE DE DATOS.....	35
Enlace de descarga .....	47

# ACERCA DE ESTE MANUAL



En este manual se describirá las funciones utilizadas en Netbeans 8.2 para poder hacer el sistema de la solución deseada:

El problema que se trata de resolver es el siguiente:

La Carnicería “La mexicana” se enfrenta a un problema de inventarios, los cuales aún su sistema es muy rustico y antiecológico usando hojas en vez de digitalizar sus reportes.

# ANTES DE EMPEZAR



## INSTALACIÓN DE MySQL Workbench Y NETBEANS8.2

Antes de empezar, debe tener un sistema operativo compatible con Microsoft Windows instalado en su computadora y debe tener un conocimiento básico de ese sistema.

Para poder proceder el programa se necesita la instalación de Netbeans 8.2 y tener el conocimiento básico.

Para esto es importante tener en cuenta el Manual de Usuario, donde se explican conceptos básicos y procesos de instalación.

# NETBEANS

## IDE 8.2



NetBeans IDE es un entorno de desarrollo integrado, gratuito y de código abierto para el desarrollo de aplicaciones.

NetBeans IDE ofrece herramientas de primera clase para el desarrollo de aplicaciones web, corporativas, de escritorio y móviles con Java. Siempre es el primer IDE en ofrecer soporte para las últimas versiones de JDK, Java EE y JavaFX. Proporciona descripciones generales inteligentes para ayudarle a comprender y gestionar sus aplicaciones.

NetBeans IDE contiene tecnologías innovadoras listas para usar y es el estándar en el desarrollo de aplicaciones, gracias a sus características integrales para el desarrollo de aplicaciones, las constantes mejoras en el editor de Java y el perfeccionamiento del rendimiento y la velocidad.

A continuación se mostrará brevemente una explicación del programa.





# CARPETA la.mexicana

## class LaMexicana

En esta clase solamente se crea la conexión a la base de datos en MySQL.

```

    * @author villa_xaz9a7o
    */
    public class LaMexicana {
        private final String base = "Lamexicana";
        private final String user = "root";
        private final String password = "123456789";
        private final String url = "jdbc:mysql://localhost:3306/"+base;
        private Connection con = null;

        public Connection getConexion() {
            try{
                Class.forName("com.mysql.cj.jdbc.Driver");
                con = (Connection) DriverManager.getConnection(this.url,this.user,this.password);
            } catch (SQLException ex) {
                System.err.print(ex);
            } catch (ClassNotFoundException e){
                Logger.getLogger(LaMexicana.class.getName()).log(Level.SEVERE,null,e);
            } return con;
        }

        public void cerrar_conexion() throws SQLException{
            if (con!=null){
                con.close();
            }
        }
    }
}
```

## Panel 1

En este JForm Solamente es de loguearse y el único botón hace la entrada a el siguiente JForm para elegir:

```

    */
    public Panel1() {
        initComponents();
        logo1.setIcon(seticon("/imagenes/logo2.jpg",logo1));
        logo2.setIcon(seticon("/imagenes/logo3.jpg",logo2));
        Color c = new Color(255,255,255);
        this.getContentPane().setBackground(c);
        setTitle("Super y Carniceria La Mexicana");
    }
}
```

Esta parte del código solo carga los componentes y los logos de la carnicería, y pone un título.

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    String usuario = "", contraseña="";  
    usuario = txt_usuario.getText();  
    contraseña = txt_contraseña.getText();  
    if (usuario.equals("Admin") && contraseña.equals("admin")) {  
        menu2 menu = new menu2();  
        menu.setVisible(true);  
  
        Panell frame2 = new Panell();  
        dispose();  
    }  
    else  
        JOptionPane.showMessageDialog(null, "Error al ingresar la informacion del usuario");  
}
```

Evento del botón para validación de usuario.

```
private void txt_usuarioMousePressed(java.awt.event.MouseEvent evt) {  
    if (txt_usuario.getText().equals("Ingrese el usuario")) {  
        txt_usuario.setText("");  
        txt_usuario.setForeground(Color.black);  
    }  
    if (String.valueOf(txt_contraseña.getPassword()).isEmpty()) {  
        txt_contraseña.setText("*****");  
        txt_contraseña.setForeground(Color.gray);  
    }  
}  
  
private void txt_contraseñaMousePressed(java.awt.event.MouseEvent evt) {  
    if (String.valueOf(txt_contraseña.getPassword()).equals("*****")) {  
        txt_contraseña.setText("");  
        txt_contraseña.setForeground(Color.gray);  
    }  
    if (txt_usuario.getText().isEmpty()) {  
        txt_usuario.setText("Ingrese el usuario");  
        txt_usuario.setForeground(Color.black);  
    }  
}
```

Formato para el campo usuario y contraseña.

```
public Icon seticon(String url, JLabel boton){  
    ImageIcon icon = new ImageIcon(getClass().getResource(url));  
    int ancho = boton.getWidth();  
    int alto = boton.getHeight();  
    ImageIcon icono = new ImageIcon(icon.getImage().getScaledInstance(ancho, alto, Image.SCALE_DEFAULT));  
    return icono;  
}
```

Formato para agregar una imagen de icono.





## MENU2

En este JForm solo son 3 botones que ayudan a cargar los siguientes JForm

```
public menu2() {  
    initComponents();  
    Agregar.setIcon(seticon("/imagenes/agregar.png", Agregar));  
    eliminar.setIcon(seticon("/imagenes/eliminar.png", eliminar));  
    Modificar.setIcon(seticon("/imagenes/editar.png", Modificar));  
}
```

Se cargan las imágenes en los botones y los componentes

```
private void AgregarActionPerformed(java.awt.event.ActionEvent evt) {  
    dispose();  
    menuagg menu = new menuagg();  
    menu.setVisible(true);  
}  
  
private void ModificarActionPerformed(java.awt.event.ActionEvent evt) {  
    dispose();  
    menu menu = new menu();  
    menu.setVisible(true);  
}  
  
private void jButton3MouseClicked(java.awt.event.MouseEvent evt) {  
    dispose();  
    Panell panelp = new Panell();  
    panelp.setVisible(true);  
}  
  
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
}  
  
private void eliminarActionPerformed(java.awt.event.ActionEvent evt) {  
    dispose();  
    menuelim eliminar = new menuelim();  
    eliminar.setVisible(true);  
}
```

Eventos de botones para cargar los siguientes JForms

```
public Icon seticon(String url, JLabel boton){  
    ImageIcon icon = new ImageIcon(getClass().getResource(url));  
    int ancho = boton.getWidth();  
    int alto = boton.getHeight();  
    ImageIcon icono = new ImageIcon(icon.getImage().getScaledInstance(ancho, alto, Image.SCALE_DEFAULT));  
    return icono;  
}
```

Formato para agregar una imagen de icono.

# MENU

En este JForm es para elegir una categoría la cual quiere ir para modificar el producto

```
public menu() {  
    initComponents();  
    pollo.setIcon(seticon("/imagenes/pollo.png",pollo));  
    res.setIcon(seticon("/imagenes/vaca.png",res));  
    puerco.setIcon(seticon("/imagenes/puerco.jpg",puerco));  
    pez.setIcon(seticon("/imagenes/pez.jpg",pez));  
}
```

Se cargan las imágenes de los botones y los componentes,

```
private void puercoActionPerformed(java.awt.event.ActionEvent evt) {  
    llenadopuerco puerco = new llenadopuerco();  
    puerco.setVisible(true);  
    menu menu = new menu();  
    dispose();  
}  
  
private void resActionPerformed(java.awt.event.ActionEvent evt) {  
    llenado res = new llenado();  
    res.setVisible(true);  
    menu menu = new menu();  
    dispose();  
}  
  
private void polloActionPerformed(java.awt.event.ActionEvent evt) {  
    llenadopollo pollo = new llenadopollo();  
    pollo.setVisible(true);  
    menu menu = new menu();  
    dispose();  
}  
  
private void pezActionPerformed(java.awt.event.ActionEvent evt) {  
    llenadopez pez = new llenadopez();  
    pez.setVisible(true);  
    menu menu = new menu();  
    dispose();  
}  
  
private void jButton3MouseClicked(java.awt.event.MouseEvent evt) {  
    dispose();  
    Panell panelp = new Panell();  
    panelp.setVisible(true);  
}  
  
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {  
    dispose();  
    menu2 menu = new menu2();  
    menu.setVisible(true);  
}
```

Código para cargar los siguientes JForms para modificar producto.

```

public Icon seticon(String url, JButton boton){
    ImageIcon icon = new ImageIcon(getClass().getResource(url));
    int ancho = boton.getWidth();
    int alto = boton.getHeight();
    ImageIcon icono = new ImageIcon(icon.getImage().getScaledInstance(ancho, alto, Image.SCALE_DEFAULT));
    return icono;
}

```

Código para cargar las imágenes en los botones.

```

class Imagenfondo2 extends JPanel{
    private Image img;
    public void paint(Graphics g){
        img = new ImageIcon(getClass().getResource("fondoll.jpg")).getImage();
        g.drawImage(img, 0, 0, getWidth(),getHeight(), this);
        setOpaque(false);
        super.paint(g);
    }
}

```

Con este método se carga una imagen de fondo para el JForm

## LLENADO

Este JForm es para modificar el producto, en este caso res

```

public llenado() {
    initComponents();
    this.setExtendedState(6);
    cargarcombonombre(combonom);
    txt_fecha.setText(fechaActual());
}

public static String fechaActual(){
    Date fecha = new Date();
    SimpleDateFormat formato = new SimpleDateFormat("YYYY/MM/dd");
    return formato.format(fecha);
}

```

Se cargan los componentes, se llama a la clase cargar combo para cargar un combobox, y se establece la fecha con el método fechaActual dándole formato a la fecha.

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    if (!"".equals(txt_id.getText())) {
        PreparedStatement ps = null;
        String id = "", entro = "", merma = "", ventas = "";
        try{
            LaMexicana objCon = new LaMexicana();
            Connection conn = objCon.getConexion();
            id = txt_id.getText();
            entro = txt_entro.getText();
            merma = txt_merma.getText();
            ventas = txt_venta.getText();

            String sql = "CALL edit_res('"+id+"', '"+entro+"', '"+merma+"', '"+ventas+"')";
            ps = conn.prepareStatement(sql);
            ps.execute();
            JOptionPane.showMessageDialog(null, "Informacion agregada correctamente");
        }catch (SQLException ex){
            JOptionPane.showMessageDialog(null, "Error al ingresar la informacion del usuario");
            System.out.println(ex);
        }
    }else{
        JOptionPane.showMessageDialog(null, "Error al ingresar la informacion del usuario");
    }
    txt_merma.setText("");
    txt_entro.setText("");
    txt_venta.setText("");
}

```

Este es el botón de agregar donde se crea la conexión, se saca el texto de los txt y luego con los valores se manda a llamar a un procedure de MySQL si se agrega correctamente manda mensaje pero si no marca error y se limpian los campos.

```

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    Document doc = new Document();

    try{
        String ruta = System.getProperty("user.home");
        PdfWriter.getInstance(doc, new FileOutputStream(ruta + "/Downloads/Reporte_Res.pdf"));
        doc.open();

        PdfPTable tabla = new PdfPTable(9);
        tabla.addCell("ID");
        tabla.addCell("PRODUCTO");
        tabla.addCell("PRECIO");
        tabla.addCell("INICIO");
        tabla.addCell("ENTRO CARNE");
        tabla.addCell("MERMA");
        tabla.addCell("VENTA");
        tabla.addCell("DISPONIBLE");
        tabla.addCell("TOTAL VENTAS");

        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = (Connection) DriverManager.getConnection("jdbc:mysql://localhost/Lamexicana", "root", " ");
            PreparedStatement pat = con.prepareStatement("select * from res;");
            ResultSet re = pat.executeQuery();

            if (re.next()) {
                do{
                    tabla.addCell(re.getString(1));
                    tabla.addCell(re.getString(2));
                    tabla.addCell(re.getString(3));
                    tabla.addCell(re.getString(4));
                    tabla.addCell(re.getString(5));
                    tabla.addCell(re.getString(6));
                }while(re.next());
            }
        }catch (Exception ex){
            JOptionPane.showMessageDialog(null, "Error al generar el reporte");
        }
    }
}

```

```

        tabla.addCell(re.getString(3));
        tabla.addCell(re.getString(4));
        tabla.addCell(re.getString(5));
        tabla.addCell(re.getString(6));
        tabla.addCell(re.getString(7));
        tabla.addCell(re.getString(8));
        tabla.addCell(re.getString(9));
    }while(re.next());
    doc.add(tabla);

    }
} catch (SQLException ex) {
    System.err.print(ex);
} catch (ClassNotFoundException e){
    Logger.getLogger(LaMexicana.class.getName()).log(Level.SEVERE,null,e);
}

doc.close();
JOptionPane.showMessageDialog(null,"Reporte creado");
} catch (DocumentException | HeadlessException | FileNotFoundException e){

}

}
}

```

Aquí se crea el documento PDF gracias a una librería donde se agregan los campos y se agrega la ruta a guardarlo, se ponen los campos a imprimir y se agregan, si no se crea correctamente el PDF se manda un mensaje de error.

```

private void txt_idKeyTyped(java.awt.event.KeyEvent evt) {
    char c = evt.getKeyChar();
    if((c<'0' || c>'9') && (c<'.' || c>'.')) evt.consume();

```

```

}

```

```

private void txt_precioKeyTyped(java.awt.event.KeyEvent evt) {
    char c = evt.getKeyChar();
    if((c<'0' || c>'9') && (c<'.' || c>'.')) evt.consume();
}

```

```

private void txt_inicioKeyTyped(java.awt.event.KeyEvent evt) {
    char c = evt.getKeyChar();
    if((c<'0' || c>'9') && (c<'.' || c>'.')) evt.consume();
}

```

```

private void txt_entroKeyTyped(java.awt.event.KeyEvent evt) {
    char c = evt.getKeyChar();
    if((c<'0' || c>'9') && (c<'.' || c>'.')) evt.consume();
}

```

```

private void txt_mermaKeyTyped(java.awt.event.KeyEvent evt) {
    char c = evt.getKeyChar();
    if((c<'0' || c>'9') && (c<'.' || c>'.')) evt.consume();
}

```

```

private void txt_ventaKeyTyped(java.awt.event.KeyEvent evt) {
    char c = evt.getKeyChar();
    if((c<'0' || c>'9') && (c<'.' || c>'.')) evt.consume();
}

```

```

private void txt_disponibleKeyTyped(java.awt.event.KeyEvent evt) {
    char c = evt.getKeyChar();
    if((c<'0' || c>'9') && (c<'.' || c>'.')) evt.consume();
}

```



Se limita a solo números decimales la entrada de datos en los txt.

```
private void combonomActionPerformed(java.awt.event.ActionEvent evt) {  
    LaMexicana objcon = new LaMexicana();  
    Connection conn = objcon.getConexion();  
  
    try {  
        PreparedStatement ps = null;  
        ResultSet rs = null;  
        ps = conn.prepareStatement("Select * FROM res where nombre = ? ");  
        ps.setString(1, (String) combonom.getSelectedItem());  
        rs = ps.executeQuery();  
        if (rs.next()) {  
            txt_id.setText(rs.getString("dni"));  
            txt_precio.setText(rs.getString("precio"));  
            txt_merma.setText("");  
            txt_entro.setText("");  
            txt_venta.setText("");  
            txt_inicio.setText(rs.getString("stock"));  
            txt_disponible.setText(rs.getString("result"));  
        }  
    } catch (SQLException ex) {  
        Logger.getLogger(llenadopuerco.class.getName()).log(Level.SEVERE, null, ex);  
    }  
}
```

Se hace que el combobox haga una consulta cada que selecciona un producto excepto los campos a modificar.

```
class Imagenfondo2 extends JPanel{  
    private Image img;  
    public void paint(Graphics g){  
        img = new ImageIcon(getClass().getResource("fondoll.jpg")).getImage();  
        g.drawImage(img, 0, 0, getWidth(),getHeight(), this);  
        setOpaque(false);  
        super.paint(g);  
    }  
}
```

Se carga una imagen de fondo

```

private void cargarcombonombre(JComboBox c) {
DefaultComboBoxModel combo = new DefaultComboBoxModel();
c.setModel(combo);
listadores listadoc = new listadores();
LaMexicana objcon = new LaMexicana();
Connection conn = objcon.getConexion();
try{
    PreparedStatement ps = null;
    ResultSet rs = null;
    String sql = "";
    sql ="CALL consultadnires()";
    ps = conn.prepareStatement(sql);
    rs = ps.executeQuery();
    while(rs.next()){
        res res = new res();
        res.setNombre(rs.getString(1));
        listadoc.agregar(res);
        combo.addElement(res.getNombre());
    }
}catch(Exception e){
    System.out.println("Error, no se puede mostrar"+e);
}
}

```

Se crea un método para cargar los productos en el combobox con una consulta para mostrar todos los campos agregándolos al combobox

Esto se repite por 3 veces más cambiando solamente las consultas en los forms:

- llenadopuerco
- llenadopez
- llenadopollo

# MENUAGG

En este JForm es para elegir una categoría la cual quiere ir para modificar el producto

```
public menuagg() {  
    initComponents();  
    pollo.setIcon(seticon("/imagenes/pollo.png",pollo));  
    res.setIcon(seticon("/imagenes/vaca.png",res));  
    puerco.setIcon(seticon("/imagenes/puerco.jpg",puerco));  
    pez.setIcon(seticon("/imagenes/pez.jpg",pez));  
}
```

Se cargan las imágenes de los botones y los componentes,

```
private void puercoActionPerformed(java.awt.event.ActionEvent evt) {  
    llenadopuerco puerco = new llenadopuerco();  
    puerco.setVisible(true);  
    menu menu = new menu();  
    dispose();  
}  
  
private void resActionPerformed(java.awt.event.ActionEvent evt) {  
    llenado res = new llenado();  
    res.setVisible(true);  
    menu menu = new menu();  
    dispose();  
}  
  
private void polloActionPerformed(java.awt.event.ActionEvent evt) {  
    llenadopollo pollo = new llenadopollo();  
    pollo.setVisible(true);  
    menu menu = new menu();  
    dispose();  
}  
  
private void pezActionPerformed(java.awt.event.ActionEvent evt) {  
    llenadopez pez = new llenadopez();  
    pez.setVisible(true);  
    menu menu = new menu();  
    dispose();  
}  
  
private void jButton3MouseClicked(java.awt.event.MouseEvent evt) {  
    dispose();  
    Panell panelp = new Panell();  
    panelp.setVisible(true);  
}  
  
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {  
    dispose();  
    menu2 menu = new menu2();  
    menu.setVisible(true);  
}
```

Código para cargar los siguientes JForms para modificar producto.

```

public Icon seticon(String url, JButton boton){
    ImageIcon icon = new ImageIcon(getClass().getResource(url));
    int ancho = boton.getWidth();
    int alto = boton.getHeight();
    ImageIcon icono = new ImageIcon(icon.getImage().getScaledInstance(ancho, alto, Image.SCALE_DEFAULT));
    return icono;
}

```

Código para cargar las imágenes en los botones.

```

class Imagenfondo2 extends JPanel{
    private Image img;
    public void paint(Graphics g){
        img = new ImageIcon(getClass().getResource("fondoll.jpg")).getImage();
        g.drawImage(img, 0, 0, getWidth(),getHeight(), this);
        setOpaque(false);
        super.paint(g);
    }
}

```

Con este método se carga una imagen de fondo para el JForm

## AGREGARRES

Se crea un JForm para agregar productos diferentes

```

public agregarres() {
    initComponents();
    this.setExtendedState(6);
    txt_fecha.setText(fechaActual());
    consulta();
}

public static String fechaActual(){
    Date fecha = new Date();
    SimpleDateFormat formato = new SimpleDateFormat("YYYY/MM/dd");
    return formato.format(fecha);
}

```

Se cargan los componentes y se agrega la fecha al txt, también se hace una consulta para mostrarla en un JTable

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    if (!"".equals(txt_id.getText())) {
        PreparedStatement ps = null;
        String producto = "", id = "", precio = "", inicio = "";
        try{
            LaMexicana objCon = new LaMexicana();
            Connection conn = objCon.getConexion();
            id = txt_id.getText();
            producto = txt_producto.getText();
            precio = txt_precio.getText();
            inicio = txt_inicio.getText();

            String sql = "CALL alta_res('"+id+"','"+producto+"','"+precio+"','"+inicio+"')";
            ps = conn.prepareStatement(sql);
            ps.execute();
            JOptionPane.showMessageDialog(null, "Informacion agregada correctamente");
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(null, "Error al ingresar la informacion del usuario");
            System.out.println(ex);
        }
    } else{
        JOptionPane.showMessageDialog(null, "Error al ingresar la informacion del usuario");
    }
    txt_id.setText("");
    txt_producto.setText("");
    txt_precio.setText("");
    txt_inicio.setText("");
}

```

Se crea la conexión con la base de datos y se saca el texto de los txt así mandando a llamar un procedimiento que hace la inserción de los datos ingresados, al no ingresarlos correctamente manda un mensaje de error al igual que si los manda correctamente manda un mensaje y al final se limpian los campos.



```

private void txt_inicioKeyTyped(java.awt.event.KeyEvent evt) {
    char c = evt.getKeyChar();
    if((c<'0' || c>'9') && (c<'.' || c>'.')) evt.consume();
}

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

private void txt_precioKeyTyped(java.awt.event.KeyEvent evt) {
    char c = evt.getKeyChar();
    if((c<'0' || c>'9') && (c<'.' || c>'.')) evt.consume();
}

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

private void txt_idKeyTyped(java.awt.event.KeyEvent evt) {
    char c = evt.getKeyChar();
    if(c<'0' || c>'9') evt.consume();
}

```

Se limita la entrada a números y decimales.

```

class Imagenfondo2 extends JPanel{
    private Image img;
    public void paint(Graphics g){
        img = new ImageIcon(getClass().getResource("fondoll.jpg")).getImage();
        g.drawImage(img, 0, 0, getWidth(), getHeight(), this);
        setOpaque(false);
        super.paint(g);
    }
}

```

Se carga una imagen de fondo

```

public void consulta() {
    String campo = txt_id.getText();
    try{
        DefaultTableModel modelo = new DefaultTableModel();
        jTable1.setModel(modelo);
        PreparedStatement ps = null;
        ResultSet rs = null;
        LaMexicana conn = new LaMexicana();
        Connection con = conn.getConexion();
        String sql = "";
        if ("".equals(campo)) {
            sql ="select dni,nombre,precio,stock from res;";
        }else{
            JOptionPane.showMessageDialog(null, "Favor de limpiar los campos");
        }
        ps = con.prepareStatement(sql);
        rs = ps.executeQuery();

        ResultSetMetaData rsMd =(ResultSetMetaData) rs.getMetaData();
        int cantidadColumnas = rsMd.getColumnCount();
        modelo.addColumn("Id Producto");
        modelo.addColumn("Producto");
        modelo.addColumn("Precio");
        modelo.addColumn("Stock");
        int[] anchos ={50,200,50,50};
        for (int i = 0; i < jTable1.getColumnCount() ; i++) {
            jTable1.getColumnModel().getColumn(i).setPreferredWidth(anchos[i]);
        }
        while (rs.next()){
            Object[] filas = new Object[cantidadColumnas];
            for (int i=0; i < cantidadColumnas; i++){
                filas[i] = rs.getObject(i+1);
            }
            modelo.addRow(filas);
        }
    }
}

```

Se crea un método consulta el cual cargara los datos en el JTable para su mejor visualización, los cuales se hacen estableciendo la conexión en la base de datos luego con un select se selecciona el producto y se agrega a la tabla.

Esto se repite por 3 veces más cambiando solamente las consultas en los forms:

- agregarpuerco
- agregarpez
- agregarpollo

# MENUELM

En este JForm es para elegir una categoría la cual quiere ir para modificar el producto

```
public menuelim() {  
    initComponents();  
    pollo.setIcon(seticon("/imagenes/pollo.png",pollo));  
    res.setIcon(seticon("/imagenes/vaca.png",res));  
    puerco.setIcon(seticon("/imagenes/puerco.jpg",puerco));  
    pez.setIcon(seticon("/imagenes/pez.jpg",pez));  
}
```

Se cargan las imágenes de los botones y los componentes,

```
private void puercoActionPerformed(java.awt.event.ActionEvent evt) {  
    llenadopuerco puerco = new llenadopuerco();  
    puerco.setVisible(true);  
    menu menu = new menu();  
    dispose();  
}  
  
private void resActionPerformed(java.awt.event.ActionEvent evt) {  
    llenado res = new llenado();  
    res.setVisible(true);  
    menu menu = new menu();  
    dispose();  
}  
  
private void polloActionPerformed(java.awt.event.ActionEvent evt) {  
    llenadopollo pollo = new llenadopollo();  
    pollo.setVisible(true);  
    menu menu = new menu();  
    dispose();  
}  
  
private void pezActionPerformed(java.awt.event.ActionEvent evt) {  
    llenadopez pez = new llenadopez();  
    pez.setVisible(true);  
    menu menu = new menu();  
    dispose();  
}  
  
private void jButton3MouseClicked(java.awt.event.MouseEvent evt) {  
    dispose();  
    Panell panelp = new Panell();  
    panelp.setVisible(true);  
}  
  
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {  
    dispose();  
    menu2 menu = new menu2();  
    menu.setVisible(true);  
}
```

Código para cargar los siguientes JForms para modificar producto.

```

public Icon seticon(String url, JButton boton){
    ImageIcon icon = new ImageIcon(getClass().getResource(url));
    int ancho = boton.getWidth();
    int alto = boton.getHeight();
    ImageIcon icono = new ImageIcon(icon.getImage().getScaledInstance(ancho, alto, Image.SCALE_DEFAULT));
    return icono;
}

```

Código para cargar las imágenes en los botones.

```

class Imagenfondo2 extends JPanel{
    private Image img;
    public void paint(Graphics g){
        img = new ImageIcon(getClass().getResource("fondoll.jpg")).getImage();
        g.drawImage(img, 0, 0, getWidth(),getHeight(), this);
        setOpaque(false);
        super.paint(g);
    }
}

```

Con este método se carga una imagen de fondo para el JForm

## ELIMINARRES

Con este método se van a eliminar los productos

```

public class eliminarres extends javax.swing.JFrame {

    public eliminarres() {
        initComponents();
        mostraardatos();
    }
}

```

Se cargan los componentes y se llama a un método el cual muestra los datos en un Jtable

```

public void mostraardatos() {
    DefaultTableModel modelo = new DefaultTableModel();
    modelo.addColumn("Dni");
    modelo.addColumn("Nombre");
    modelo.addColumn("Precio");
    modelo.addColumn("Entro");
    modelo.addColumn("Merma");
    modelo.addColumn("Vendido");
    modelo.addColumn("Inicio");
    modelo.addColumn("Disponible");

    tablaelim.setModel(modelo);
    String[] datos = new String[8];

    try {
        PreparedStatement ps = null;
        ResultSet rs = null;
        LaMexicana conn = new LaMexicana();
        Connection con = conn.getConexion();
        String sql = "";
        sql = "Select * from res";
        ps = con.prepareStatement(sql);
        rs = ps.executeQuery();
        while(rs.next()){
            datos[0] = rs.getString(1);
            datos[1] = rs.getString(2);
            datos[2] = rs.getString(3);
            datos[3] = rs.getString(4);
            datos[4] = rs.getString(5);
            datos[5] = rs.getString(6);
            datos[6] = rs.getString(7);
            datos[7] = rs.getString(8);
            modelo.addRow(datos);
        }
        tablaelim.setModel(modelo);

        tablaelim.setModel(modelo);
    } catch (SQLException ex) {
        Logger.getLogger(eliminarres.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

Con este método se agregan los campos a un Jtable el cual se carga mediante una instrucción de MySQL



```

public void eliminar() {
    int fila = tablaelim.getSelectedRow();
    String valor = tablaelim.getValueAt(fila, 0).toString();

    try {
        PreparedStatement ps = null;
        ResultSet rs = null;
        LaMexicana conn = new LaMexicana();
        Connection con = conn.getConexion();
        String sql = "";
        sql = "call elim_res ('"+valor+"')";
        ps = con.prepareStatement(sql);
        ps.execute();
        mostraardatos();
        rs = ps.executeQuery();
    } catch (SQLException ex) {
        Logger.getLogger(eliminarres.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

Con este método eliminar se elimina el producto una vez seleccionado el id del producto en la tabla donde se despliegan todos los productos, llamando a un procedure de MySQL.

Esto se repite por 3 veces mas cambiando solamente las consultas en los forms:

- eliminarpuerco
- eliminarpez
- eliminarpollo

# BASE DE DATOS

```
create database lamexicana;
```

```
use lamexicana;
```

```
create table res(  
dni int not null primary key,  
nombre varchar(255),  
precio float,  
entro float,  
merma float,  
dvendido float,  
stock float,  
result float,  
venta float  
);
```

```
create table puerco(  
dni int not null primary key,  
nombre varchar(255),  
precio float,  
merma float,  
entro float,  
dvendido float,  
stock float,  
result float,
```

venta float

);

create table pollo(

dni int not null primary key,

nombre varchar(255),

precio float,

merma float,

entro float,

dvendido float,

stock float,

result float,

venta float

);

create table pez(

dni int not null primary key,

nombre varchar(255),

precio float,

entro float,

merma float,

dvendido float,

stock float,

result float,

venta float

);

```
delimiter //  
CREATE PROCEDURE mostrar_res(in id int)  
begin  
select result from pez where dni = id;  
end//  
delimiter ;
```

```
create procedure consultadnipuerco()  
select nombre from puerco;
```

```
create procedure consultadnires()  
select nombre from res;
```

```
create procedure consultadnipollo()  
select nombre from pollo;
```

```
create procedure consultadnipez()  
select nombre from pez;
```

```
call consultadnires();
```

```
create procedure consultaridres(in id int)  
select * from res where dni = id;
```

```
create procedure consultaridpez(in id int)  
select * from pez where dni = id;
```

```
create procedure consultaridpollo(in id int)
```

```
select * from pollo where dni = id;
```

```
#Editar puerco
```

```
delimiter //
```

```
CREATE PROCEDURE edit_puerco(IN id int,IN ent float ,IN mer float,IN  
dinvend float)
```

```
begin
```

```
if exists(select id from puerco where dni like id)
```

```
then
```

```
update puerco set entro = ent, merma = mer, dvendido = dinvend, result  
= (stock+entro-merma-dvendido), venta = (dvendido*precio) where dni  
like id;
```

```
end if;
```

```
end//
```

```
delimiter ;
```

```
#Editar res
```

```
delimiter //
```

```
CREATE PROCEDURE edit_res(IN id int,IN ent float ,IN mer float,IN dinvend  
float)
```

```
begin
```

```
if exists(select id from res where dni like id)
```

```
then
```

```
update res set entro = ent, merma = mer, dvendido = dinvend, result =  
(stock+entro-merma-dvendido), venta = (dvendido*precio)where dni like  
id;
```

```
end if;
```

```
end//
```

```
delimiter ;
```



#Editar pollo

delimiter //

```
CREATE PROCEDURE edit_pollo(IN id int,IN ent float ,IN mer float,IN
dinvend float)
```

```
begin
```

```
if exists(select id from pollo where dni like id)
```

```
then
```

```
update pollo set entro = ent, merma = mer, dvendido = dinvend, result =
(stock+entro-merma-dvendido) , venta = (dvendido*precio) where dni like
id;
```

```
end if;
```

```
end//
```

```
delimiter ;
```

#Editar pez

delimiter //

```
CREATE PROCEDURE edit_pez(IN id int,IN ent float ,IN mer float,IN dinvend
float)
```

```
begin
```

```
if exists(select id from pez where dni like id)
```

```
then
```

```
update pez set entro = ent, merma = mer, dvendido = dinvend, result =
(stock+entro-merma-dvendido), venta = (dvendido*precio) where dni like
id;
```

```
end if;
```

```
end//
```

```
delimiter ;
```

#Alta pollo

delimiter //

```
CREATE PROCEDURE alta_pollo(IN id int, IN nom varchar(255), IN pre float,  
IN stok float)
```

```
begin
```

```
if not exists(select id from pollo where nombre like nom)
```

```
then
```

```
insert into pollo (dni,nombre,precio,stock) values (id,nom,pre,stok);
```

```
end if;
```

```
end//
```

```
delimiter ;
```

#Alta puerco

delimiter //

```
CREATE PROCEDURE alta_puerco(IN id int, IN nom varchar(255), IN pre  
float, IN stok float)
```

```
begin
```

```
if not exists(select id from puerco where nombre like nom)
```

```
then
```

```
insert into puerco (dni,nombre,precio,stock) values (id,nom,pre,stok);
```

```
end if;
```

```
end//
```

```
delimiter ;
```

#Alta res

delimiter //

```
CREATE PROCEDURE alta_res(IN id int, IN nom varchar(255), IN pre float, IN  
stok float)
```

```
begin
```

```
if not exists(select id from res where nombre like nom)
```

```

then
insert into res (dni,nombre,precio,stock) values (id,nom,pre,stok);
end if;
end//
delimiter ;

#Alta pez
delimiter //
CREATE PROCEDURE alta_pez(IN id int, IN nom varchar(255), IN pre float,
IN stok float)
begin
if not exists(select id from pez where nombre like nom)
then
insert into pez (dni,nombre,precio,stock) values (id,nom,pre,stok);
end if;
end//
delimiter ;

select dni,nombre,precio,stock from res;

#Eliminar pollo
delimiter //
create procedure elim_pollo(in id int)
begin
if exists (select dni from pollo where dni like id)
then
    delete from pollo where dni like id;
end if;
end//

```

delimiter ;

#Eliminar pez

delimiter //

create procedure elim\_pez(in id int)

begin

if exists (select dni from pez where dni like id)

then

delete from pez where dni like id;

end if;

end//

delimiter ;

#Eliminar puerco

delimiter //

create procedure elim\_puerco(in id int)

begin

if exists (select dni from puerco where dni like id)

then

delete from puerco where dni like id;

end if;

end//

delimiter ;

#Eliminar res

delimiter //

create procedure elim\_res(in id int)

begin

```
if exists (select dni from res where dni like id)
then
    delete from res where dni like id;
end if;
end//
delimiter ;
```

#Backup

```
create table resb(
dni int,
nombre varchar(255),
precio float,
entro float,
merma float,
dvendido float,
stock float,
result float,
venta float
);
```

#Por si acaso res

```
delimiter //
create trigger resdeleted after delete on res for each row
begin
    insert into resb
(dni,nombre,precio,entro,merma,dvendido,stock,result,venta) values
(old.dni,old.nombre,old.precio,old.entro,old.merma,old.dvendido,old.stock
,old.result,old.venta);
end//
```

delimiter ;

#Backup

create table puercob(

dni int,

nombre varchar(255),

precio float,

entro float,

merma float,

dvendido float,

stock float,

result float,

venta float

);

#Por si acaso puerco

delimiter //

create trigger puercodeleted after delete on res for each row

begin

insert into puercob

(dni,nombre,precio,entro,merma,dvendido,stock,result,venta) values

(old.dni,old.nombre,old.precio,old.entro,old.merma,old.dvendido,old.stock  
,old.result,old.venta);

end//

delimiter ;

#Backup

create table pezb(

dni int,

```
nombre varchar(255),  
precio float,  
entro float,  
merma float,  
dvendido float,  
stock float,  
result float,  
venta float  
);
```

```
#Por si acaso pez
```

```
delimiter //
```

```
create trigger pezdeleted after delete on res for each row
```

```
begin
```

```
insert into pezb
```

```
(dni,nombre,precio,entro,merma,dvendido,stock,result,venta) values
```

```
(old.dni,old.nombre,old.precio,old.entro,old.merma,old.dvendido,old.stock  
,old.result,old.venta);
```

```
end//
```

```
delimiter ;
```

```
#Backup
```

```
create table pollob(
```

```
dni int,
```

```
nombre varchar(255),
```

```
precio float,
```

```
entro float,
```

```
merma float,
```

```
dvendido float,
```



```

stock float,
result float,
venta float
);

#Por si acaso res
delimiter //
create trigger polodeleted after delete on res for each row
begin
    insert into pollob
(dni,nombre,precio,entro,merma,dvendido,stock,result,venta) values
(old.dni,old.nombre,old.precio,old.entro,old.merma,old.dvendido,old.stock
,old.result,old.venta);
end//
delimiter ;

create table reportesres(
dni_res int not null,
fecha date,
venta float,
constraint Fk_dnires foreign key (dni_res) references res (dni)
);

insert into res (dni,nombre,precio,entro,merma,dvendido,stock,result)
values (1,"Chuleta",100,100,20,20,50,(stock+entro-merma-dvendido))

#Alta res
delimiter //
CREATE PROCEDURE venta_res(IN id int, in fech date)

```

```
begin
if exists(select id from res where dni like id)
then
insert into reportesres (dni_res,fecha) values (id,fech);
end if;
end//
delimiter ;
```

```
INSERT INTO reportesres( venta )
SELECT venta FROM res
```

## Enlace de descarga

[https://drive.google.com/drive/folders/1cGR\\_JadDFB-BVogV650OKzHb7jMgH-Sd?usp=share\\_link](https://drive.google.com/drive/folders/1cGR_JadDFB-BVogV650OKzHb7jMgH-Sd?usp=share_link)