Aplicación informática en procesos de producción de tilapias (Oreochromis niloticus) (TPI)

Clase ingreso

```
import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.table.DefaultTableModel;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.GroupLayout;
import javax.swing.GroupLayout.Alignment;
import javax.swing.JTextField;
import javax.swing.JButton;
import javax.swing.LayoutStyle.ComponentPlacement;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JDesktopPane;

public class FrmIngreso extends JFrame {

	private JPanel contentPane;
	private JTextField txtUsuario;
	private JTextField txtClave;

	Conexion conexion = new Conexion();
	Connection cn=null;
	Statement stm=null;
	ResultSet rs=null;

	public boolean Encontrado;

	void tmpUser(String nombUser, String clavUser)
	{
		try
		{
			cn = conexion.Conectar();
			stm =cn.createStatement();
			rs=stm.executeQuery("SELECT * FROM tblusuarios");

			Encontrado=false;

			while(rs.next())
				{
```

```
                                        String strNombre = rs.getString(2);
                                        String strClave = rs.getString(3);

                                        if ((strNombre.equals(nombUser)) &&
(strClave.equals(clavUser)))
                                                Encontrado=true;
                                }
                        }
                        catch (SQLException e)
                        {
                                e.printStackTrace();
                        }
                        finally
                        {
            try
            {
               if (rs!= null)
               {
                  rs.close();
               }

               if (stm != null)
               {
                  stm.close();
               }
               if (cn != null)
               {
                  cn.close();
               }
            }
            catch (Exception e2)
            {
               e2.printStackTrace();
            }
         }
           }

           /**
            * Launch the application.
            */
           public static void main(String[] args) {
                   EventQueue.invokeLater(new Runnable() {
                           public void run() {
                                   try {
                                           FrmIngreso frame = new FrmIngreso();
                                           frame.setVisible(true);
                                   } catch (Exception e) {
                                           e.printStackTrace();
                                   }
                           }
                   });
           }

           /**
            * Create the frame.
            */
           public FrmIngreso() {
                   setResizable(false);
                   setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                   setBounds(100, 100, 349, 158);
                   contentPane = new JPanel();
                   contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
```

```
		setContentPane(contentPane);

		JLabel lblUsuario = new JLabel("Usuario");

		txtUsuario = new JTextField();
		txtUsuario.setColumns(10);

		JLabel lblClave = new JLabel("Clave");

		txtClave = new JTextField();
		txtClave.setColumns(10);

		JButton btnIniciar = new JButton("Iniciar");
		btnIniciar.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent arg0) {
			}
		});
		btnIniciar.addMouseListener(new MouseAdapter() {
			@Override
			public void mouseClicked(MouseEvent arg0)
			{
				String strUsuario = txtUsuario.getText();
				String strClave = txtClave.getText();

				tmpUser(strUsuario, strClave);
				if (Encontrado==true)
				{
					JOptionPane.showMessageDialog(null, "Acceso correcto");

					FrmPrincipal vntPrincipal = new FrmPrincipal();
					//panelinter.add(vntTempe);
					vntPrincipal.setVisible(true);
					dispose();
				}
				else
					JOptionPane.showMessageDialog(null, "Usuario o clave
incorrecta");
			}
		});

		JButton btnSalir = new JButton("Salir");
		btnSalir.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
			}
		});
		btnSalir.addMouseListener(new MouseAdapter() {
			@Override
			public void mouseClicked(MouseEvent e)
			{
				System.exit(0);
			}
		});

		JDesktopPane desktopPane = new JDesktopPane();

		JButton btnNewButton = new JButton("New button");

		JButton btnNewButton_1 = new JButton("New button");
		GroupLayout gl_contentPane = new GroupLayout(contentPane);
		gl_contentPane.setHorizontalGroup(
			gl_contentPane.createParallelGroup(Alignment.LEADING)
				.addGroup(gl_contentPane.createSequentialGroup()
```

```
                                        .addGap(43)

                .addGroup(gl_contentPane.createParallelGroup(Alignment.TRAILING)
                                                .addComponent(lblClave)
                                                .addComponent(lblUsuario))
                                        .addGap(31)

                .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
                                                .addGroup(gl_contentPane.createSequentialGroup()
                                                        .addComponent(btnNewButton)

                .addPreferredGap(ComponentPlacement.RELATED)
                                                        .addComponent(desktopPane,
GroupLayout.PREFERRED_SIZE, 1, GroupLayout.PREFERRED_SIZE)

                .addPreferredGap(ComponentPlacement.RELATED, 28, Short.MAX_VALUE)
                                                        .addComponent(btnNewButton_1))

                .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING, false)

                .addGroup(gl_contentPane.createSequentialGroup()
                                                                .addComponent(txtUsuario,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                                                .addGap(18)
                                                                .addComponent(btnIniciar))

                .addGroup(gl_contentPane.createSequentialGroup()
                                                                .addComponent(txtClave,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                                                .addGap(18)
                                                                .addComponent(btnSalir,
GroupLayout.DEFAULT_SIZE, GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))))
                                        .addContainerGap())
                );
                gl_contentPane.setVerticalGroup(
                        gl_contentPane.createParallelGroup(Alignment.LEADING)
                                .addGroup(gl_contentPane.createSequentialGroup()
                                        .addGap(23)

                .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                                .addComponent(lblUsuario)
                                                .addComponent(txtUsuario,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                                .addComponent(btnIniciar))
                                        .addPreferredGap(ComponentPlacement.UNRELATED)

                .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                                .addComponent(lblClave)
                                                .addComponent(txtClave,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                                .addComponent(btnSalir))
                                        .addPreferredGap(ComponentPlacement.RELATED,
GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                .addGroup(gl_contentPane.createParallelGroup(Alignment.TRAILING)
                                                .addComponent(desktopPane,
GroupLayout.PREFERRED_SIZE, 1, GroupLayout.PREFERRED_SIZE)

                .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                                        .addComponent(btnNewButton)
                                                        .addComponent(btnNewButton_1)))
                                        .addContainerGap())
```

```
                );
                contentPane.setLayout(gl_contentPane);
        }
}
```

Clase Principal

```
import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JMenuBar;
import javax.swing.JMenu;
import javax.swing.JMenuItem;
import javax.swing.GroupLayout;
import javax.swing.GroupLayout.Alignment;
import javax.swing.JDesktopPane;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class FrmPrincipal extends JFrame {

        private JPanel contentPane;

        /**
         * Launch the application.
         */
        public static void main(String[] args) {
                EventQueue.invokeLater(new Runnable() {
                        public void run() {
                                try {
                                        FrmPrincipal frame = new FrmPrincipal();
                                        frame.setVisible(true);
                                } catch (Exception e) {
                                        e.printStackTrace();
                                }
                        }
                });
        }

        /**
         * Create the frame.
         */
        public FrmPrincipal() {
                JDesktopPane panelinter = new JDesktopPane();
                panelinter.addMouseListener(new MouseAdapter() {
                        @Override
                        public void mouseClicked(MouseEvent e) {
                        }
                });

                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                setBounds(100, 100, 646, 399);
```

```
		JMenuBar menuBar = new JMenuBar();
		setJMenuBar(menuBar);

		JMenu mnControl = new JMenu("Control");
		menuBar.add(mnControl);

		JMenuItem mntmTemperatura = new JMenuItem("Temperatura");
		mntmTemperatura.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
				FrmTempe vntTempe = new FrmTempe();
				//panelinter.add(vntTempe);
				vntTempe.setVisible(true);
			}
		});
		mnControl.add(mntmTemperatura);

		JMenuItem mntmUsuario = new JMenuItem("Usuario");
		mntmUsuario.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent arg0) {
				FrmUsuario vntUsuario = new FrmUsuario();
				//panelinter.add(vntUsuario);
				vntUsuario.setVisible(true);
			}
		});
		mnControl.add(mntmUsuario);

		JMenuItem mntmSimulacin = new JMenuItem("Simulaci\u00F3n");
		mntmSimulacin.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
				FrmSimulacion vntSimulacion = new FrmSimulacion();
				//panelinter.add(vntSimulacion);
				vntSimulacion.setVisible(true);
			}
		});
		mnControl.add(mntmSimulacin);

		JMenuItem mntmSalir = new JMenuItem("Salir");
		mntmSalir.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent arg0) {
				System.exit(0);
			}
		});
		mnControl.add(mntmSalir);
		contentPane = new JPanel();
		contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
		setContentPane(contentPane);


		GroupLayout gl_contentPane = new GroupLayout(contentPane);
		gl_contentPane.setHorizontalGroup(
			gl_contentPane.createParallelGroup(Alignment.LEADING)
				.addComponent(panelinter, GroupLayout.DEFAULT_SIZE, 630,
Short.MAX_VALUE)
		);
		gl_contentPane.setVerticalGroup(
			gl_contentPane.createParallelGroup(Alignment.LEADING)
				.addComponent(panelinter, GroupLayout.DEFAULT_SIZE, 337,
Short.MAX_VALUE)
		);
		contentPane.setLayout(gl_contentPane);
	}
```

}

Clase Usuario

```
//import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.GroupLayout;
import javax.swing.GroupLayout.Alignment;
import javax.swing.JButton;
import javax.swing.LayoutStyle.ComponentPlacement;
//import javax.swing.SwingConstants;
//import java.awt.Component;
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import javax.swing.JOptionPane;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;


public class FrmUsuario extends JFrame {

	private JPanel contentPane;
	private JTextField txtNombUsuario;
	private JTextField txtClavUsuario;
	private JTable tbluser;

	Conexion conexion = new Conexion();
	Connection cn=null;
	Statement stm=null;
	ResultSet rs=null;

	public int codUMax =0;
	private JTextField txtCodUsuario;

	void mostrarRegistros()
	{
		try
		{
			cn = conexion.Conectar();
			stm =cn.createStatement();
			rs=stm.executeQuery("SELECT * FROM tblusuarios");

			int numCols = tbluser.getModel().getColumnCount();
			if (numCols>0)
			{
				while(rs.next())
				{
```

```
                                        Object [] filtbluser = new Object[numCols];
                                        if (rs.getInt(1)>codUMax)
                                        {
                                                codUMax=rs.getInt(1);
                                        }
                                        filtbluser[0] = rs.getInt(1);
                                        filtbluser[1] = rs.getString(2);
                                        filtbluser[2] = rs.getString(3);
                                        ((DefaultTableModel) tbluser.getModel()).addRow(filtbluser);
                                }
                        }
                        else
                        {
                                JOptionPane.showMessageDialog(null, "La tabla se encuentra vacía");
                        }

                }
                catch (SQLException e)
                {
                        e.printStackTrace();
                }
                finally
                {
    try
    {
        if (rs!= null)
        {
            rs.close();
        }

        if (stm != null)
        {
            stm.close();
        }
        if (cn != null)
        {
            cn.close();
        }
    }
    catch (Exception e2)
    {
        e2.printStackTrace();
    }
}
     }

     void mostrarInfo(int posFil)
     {
                txtCodUsuario.setText(String.valueOf(tbluser.getValueAt(posFil,0)));
                txtNombUsuario.setText(String.valueOf(tbluser.getValueAt(posFil,1)));
                txtClavUsuario.setText(String.valueOf(tbluser.getValueAt(posFil,2)));
     }

     /**
      * Launch the application.
      */
     public static void main(String[] args) {
                EventQueue.invokeLater(new Runnable() {
                        public void run() {
                                try {
                                        FrmUsuario frame = new FrmUsuario();
                                        frame.setVisible(true);
```

```
                                        } catch (Exception e) {
                                                e.printStackTrace();
                                        }
                                }
                        });
        }

        /**
         * Create the frame.
         */
        public FrmUsuario() {
                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                setBounds(100, 100, 511, 351);
                contentPane = new JPanel();
                contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
                setContentPane(contentPane);

                JScrollPane scrollPane = new JScrollPane();
                JButton btnNuevo = new JButton("Nuevo");
                btnNuevo.addMouseListener(new MouseAdapter() {
                        @Override
                        public void mouseClicked(MouseEvent arg0) {
                                txtCodUsuario.setText(String.valueOf(codUMax+1));
                                txtNombUsuario.setText("");
                                txtClavUsuario.setText("");
                        }
                });

                JButton btnGuardar = new JButton("Guardar");
                btnGuardar.addMouseListener(new MouseAdapter() {
                        @Override
                        public void mouseClicked(MouseEvent arg0) {

                                try
                                {
                                        cn = conexion.Conectar();
                                        stm =cn.createStatement();

                                        String strNombUsuario = txtNombUsuario.getText();
                                        String strClavUsuario = txtClavUsuario.getText();

                                        String strSql = "INSERT INTO tblusuarios(nombusuario,
clavusuario) VALUES ('";
                                        strSql = strSql + strNombUsuario + "', '"+strClavUsuario+"')";
                                        stm.executeUpdate(strSql);

                                        tbluser.setModel(new DefaultTableModel(
                                                new Object[][] {
                                                },
                                                new String[] {
                                                                "C\u00F3digo","Usuario", "Clave"
                                                }
                                        ));
                                        scrollPane.setViewportView(tbluser);

                                        mostrarRegistros();

                                        int cantRegTbl =tbluser.getRowCount();
                                        if (cantRegTbl>1)
                                        {
                                                int posRegTbl = tbluser.getRowCount();
                                                mostrarInfo(posRegTbl);
```

```
                        }
                    }
                    catch (SQLException e3)
                    {
                        e3.printStackTrace();
                    }

                }
            });
            btnGuardar.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent arg0) {
                }
            });

            JButton btnModificar = new JButton("Modificar");
            btnModificar.addMouseListener(new MouseAdapter() {
                @Override
                public void mouseClicked(MouseEvent e) {

                    try
                    {
                        cn = conexion.Conectar();
                        stm =cn.createStatement();

                        String strCodigo = txtCodUsuario.getText();
                        String strNombUsuario = txtNombUsuario.getText();
                        String strClavUsuario = txtClavUsuario.getText();

                        String strSql = "UPDATE tblusuarios ";
                        strSql = strSql + "SET nombusuario ='"+strNombUsuario+"', ";
                        strSql = strSql + "clavusuario ='"+strClavUsuario+"' ";
                        strSql = strSql + "WHERE codusuario= "+strCodigo;
                        stm.executeUpdate(strSql);


                        tbluser.setModel(new DefaultTableModel(
                                new Object[][] {
                                },
                                new String[] {
                                        "C\u00F3digo","Usuario",
"Clave"
                                }
                            ));

                        scrollPane.setViewportView(tbluser);

                        mostrarRegistros();
                    }
                    catch (SQLException e4)
                    {
                        e4.printStackTrace();
                    }

                }
            });
            btnModificar.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                }
            });

            JButton btnEliminar = new JButton("Eliminar");
            btnEliminar.addMouseListener(new MouseAdapter() {
```

```
                @Override
                public void mouseClicked(MouseEvent e) {
                        try
                        {
                                cn = conexion.Conectar();
                                stm =cn.createStatement();

                                String strCodigo = txtCodUsuario.getText();
                                String strSql = "DELETE FROM tblusuarios WHERE codusuario
= '';
                                strSql = strSql + strCodigo;

                                stm.executeUpdate(strSql);

                                tbluser.setModel(new DefaultTableModel(
                                        new Object[][] {
                                        },
                                        new String[] {
                                                "C\u00F3digo", "Fecha", "Grados"
                                        }
                                ));
                                scrollPane.setViewportView(tbluser);

                                mostrarRegistros();
                        }
                        catch (SQLException e5)
                        {
                                e5.printStackTrace();
                        }
                }
        });

        JButton btnSalir = new JButton("Salir");
        btnSalir.addMouseListener(new MouseAdapter() {
                @Override
                public void mouseClicked(MouseEvent e) {
                        dispose();
                }
        });

        JLabel lblUsuario = new JLabel("Usuario");

        txtNombUsuario = new JTextField();
        txtNombUsuario.setColumns(10);

        txtClavUsuario = new JTextField();
        txtClavUsuario.setColumns(10);

        JLabel lblClave = new JLabel("Clave");

        JLabel lblCdigo = new JLabel("C\u00F3digo");

        txtCodUsuario = new JTextField();
        txtCodUsuario.setEnabled(false);
        txtCodUsuario.setColumns(10);
        GroupLayout gl_contentPane = new GroupLayout(contentPane);
        gl_contentPane.setHorizontalGroup(
                gl_contentPane.createParallelGroup(Alignment.LEADING)
                        .addGroup(gl_contentPane.createSequentialGroup()
                                .addGap(37)

    .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
```

```
                                                .addGroup(gl_contentPane.createSequentialGroup()
                                                        .addComponent(lblClave)
                                                        .addGap(35)
                                                        .addComponent(txtClavUsuario,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
                                                .addGroup(gl_contentPane.createSequentialGroup()

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
                                                                .addComponent(lblUsuario)
                                                                .addComponent(lblCdigo))
                                                        .addGap(26)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
                                                                .addComponent(txtCodUsuario,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                                                .addComponent(txtNombUsuario,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))))
                                                .addGap(59)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING, false)
                                                .addComponent(btnNuevo,
GroupLayout.DEFAULT_SIZE, 83, Short.MAX_VALUE)
                                                .addComponent(btnGuardar,
GroupLayout.DEFAULT_SIZE, GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                                .addComponent(btnModificar,
GroupLayout.DEFAULT_SIZE, GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                                                .addGap(18)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING, false)
                                                .addComponent(btnSalir, GroupLayout.DEFAULT_SIZE,
GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                                .addComponent(btnEliminar,
GroupLayout.DEFAULT_SIZE, 84, Short.MAX_VALUE))
                                                .addContainerGap(56, Short.MAX_VALUE))
                                        .addComponent(scrollPane, GroupLayout.DEFAULT_SIZE, 485,
Short.MAX_VALUE)
                );
                gl_contentPane.setVerticalGroup(
                        gl_contentPane.createParallelGroup(Alignment.LEADING)
                                .addGroup(gl_contentPane.createSequentialGroup()

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
                                                .addGroup(gl_contentPane.createSequentialGroup()
                                                        .addContainerGap()

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                                                .addComponent(lblCdigo)
                                                                .addComponent(txtCodUsuario,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                                                .addComponent(btnNuevo))
                                                        .addGap(7)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)

        .addGroup(gl_contentPane.createSequentialGroup()
                                                                        .addGap(3)
                                                                        .addComponent(lblUsuario))

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)

        .addComponent(txtNombUsuario, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE)
```

```
		.addComponent(btnGuardar)))

		.addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)

		.addGroup(gl_contentPane.createSequentialGroup()
                                                                                .addGap(6)

		.addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)

		.addGroup(gl_contentPane.createSequentialGroup()
                                                                                .addGap(7)

		.addComponent(lblClave))

		.addGroup(gl_contentPane.createSequentialGroup()
                                                                                .addGap(4)

		.addComponent(txtClavUsuario, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))))

		.addGroup(gl_contentPane.createSequentialGroup()

		.addPreferredGap(ComponentPlacement.UNRELATED)

		.addComponent(btnModificar))))
                                                .addGroup(gl_contentPane.createSequentialGroup()
                                                        .addGap(25)
                                                        .addComponent(btnEliminar)

		.addPreferredGap(ComponentPlacement.UNRELATED)
                                                        .addComponent(btnSalir)))
                                        .addGap(18)
                                        .addComponent(scrollPane, GroupLayout.DEFAULT_SIZE, 186,
Short.MAX_VALUE))
                );

                tbluser = new JTable();
                tbluser.addMouseListener(new MouseAdapter() {
                        @Override
                        public void mouseClicked(MouseEvent e)
                        {
                                int posregtbluser=tbluser.getSelectedRow();
                                mostrarInfo(posregtbluser);
                        }
                });
                tbluser.setModel(new DefaultTableModel(
                        new Object[][] {
                        },
                        new String[] {
                                        "C\u00F3digo","Usuario", "Clave"
                        }
                ));
                scrollPane.setViewportView(tbluser);
                contentPane.setLayout(gl_contentPane);

                mostrarRegistros();
                if (tbluser.getRowCount()>1)
                {
                        mostrarInfo(0);
                }
        }
```

}

Clase Temperatura

```
//import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.GroupLayout;
import javax.swing.GroupLayout.Alignment;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.LayoutStyle.ComponentPlacement;
import javax.swing.JTextField;
import javax.swing.JButton;
import javax.swing.SwingConstants;
import java.awt.Component;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
//import javax.swing.JFormattedTextField;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.event.FocusAdapter;
import java.awt.event.FocusEvent;

// Fechas
//import java.util.Date;
//import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.GregorianCalendar;

public class FrmTempe extends JFrame {

	private JPanel contentPane;
	private JTable tbltempe;
	private JTextField txtCodigo;
	private JTextField txtFecha;
	private JTextField txtGrados;

	Conexion conexion = new Conexion();
	Connection cn=null;
	Statement stm=null;
	ResultSet rs=null;

	public int codMax =0;

	void mostrarRegistros()
	{
```

```
                try
                {
                        cn = conexion.Conectar();
                        stm =cn.createStatement();
                        rs=stm.executeQuery("SELECT * FROM tbltemperatura");

                        int numCols = tbltempe.getModel().getColumnCount();
                        if (numCols>0)
                        {
                                while(rs.next())
                                {
                                        Object [] filtbltempe = new Object[numCols];
                                        if (rs.getInt(1)>codMax)
                                        {
                                                codMax=rs.getInt(1);
                                        }
                                        filtbltempe[0] = rs.getInt(1);
                                        filtbltempe[1] = rs.getDate(2);
                                        filtbltempe[2] = rs.getInt(3);
                                       ((DefaultTableModel) tbltempe.getModel()).addRow(filtbltempe);
                                }
                        }
                        else
                        {
                                JOptionPane.showMessageDialog(null, "La tabla se encuentra vacía");
                        }

                }
                catch (SQLException e)
                {
                        e.printStackTrace();
                }
                finally
                {
    try
    {
        if (rs!= null)
        {
            rs.close();
        }

        if (stm != null)
        {
            stm.close();
        }
        if (cn != null)
        {
            cn.close();
        }
    }
    catch (Exception e2)
    {
        e2.printStackTrace();
    }
}
     }


      void mostrarInfo(int posFil)
      {
                txtCodigo.setText(String.valueOf(tbltempe.getValueAt(posFil,0)));
                txtFecha.setText(String.valueOf(tbltempe.getValueAt(posFil,1)));
```

```
		txtGrados.setText(String.valueOf(tbltempe.getValueAt(posFil,2)));
}

/**
 * Launch the application.
 */
public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					FrmTempe frame = new FrmTempe();
					frame.setVisible(true);
				} catch (Exception e) {
					e.printStackTrace();
				}
			}
		});
}

/**
 * Create the frame.
 */
public FrmTempe() {
		setTitle("TEMPERATURAS");
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setBounds(100, 100, 552, 300);
		contentPane = new JPanel();
		contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
		setContentPane(contentPane);
		JScrollPane scrollPane = new JScrollPane();
		scrollPane.addFocusListener(new FocusAdapter() {
			@Override
			public void focusGained(FocusEvent arg0)
			{
			}
		});
		scrollPane.addKeyListener(new KeyAdapter() {
			@Override
			public void keyPressed(KeyEvent arg0)
			{
			}
			@Override
			public void keyReleased(KeyEvent e)
			{
			}
		});
		scrollPane.addMouseListener(new MouseAdapter() {
			@Override
			public void mousePressed(MouseEvent arg0)
			{
			}
		});

		JLabel lblCodigo = new JLabel("C\u00F3digo");

		JLabel lblFecha = new JLabel("Fecha");

		JLabel lblGrados = new JLabel("Grados");

		txtCodigo = new JTextField();
		txtCodigo.setEnabled(false);
		txtCodigo.setColumns(10);
```

```
		txtFecha = new JTextField();
		txtFecha.setColumns(10);

		txtGrados = new JTextField();
		txtGrados.setColumns(10);

		JButton btnLimpiar = new JButton("Nuevo");
		btnLimpiar.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent arg0) {
			}
		});
		btnLimpiar.addMouseListener(new MouseAdapter() {
			@Override
			public void mouseClicked(MouseEvent e)
			{
				String strFecha;
				Calendar fecha = new GregorianCalendar();
				int anio = fecha.get(Calendar.YEAR);
				int mes = fecha.get(Calendar.MONTH);
		int dia = fecha.get(Calendar.DAY_OF_MONTH);
		strFecha = String.valueOf(anio)+"-"+String.valueOf(mes+1)+"-"+String.valueOf(dia);

				txtCodigo.setText(String.valueOf(codMax+1));
				txtFecha.setText(strFecha);
				txtGrados.setText("");
			}
		});

		JButton btnGuardar = new JButton("Guardar");
		btnGuardar.addMouseListener(new MouseAdapter()
		{
			@Override
			public void mouseClicked(MouseEvent e)
			{
				try
				{
					cn = conexion.Conectar();
					stm =cn.createStatement();

					String strFecha = txtFecha.getText();
					String strGrados = txtGrados.getText();

					String strSql = "INSERT INTO tbltemperatura(fechtempe,
gradtempe) VALUES ('";
					strSql = strSql + strFecha + "', "+strGrados+")";
					stm.executeUpdate(strSql);

					tbltempe.setModel(new DefaultTableModel(
							new Object[][] {
							},
							new String[] {
								"C\u00F3digo", "Fecha", "Grados"
							}
						));
					scrollPane.setViewportView(tbltempe);

					mostrarRegistros();

					int cantRegTbl =tbltempe.getRowCount();
					if (cantRegTbl>1)
					{
```

```
                                        int posRegTbl = tbltempe.getRowCount();
                                        mostrarInfo(posRegTbl);
                                }
                        }
                        catch (SQLException e3)
                        {
                                e3.printStackTrace();
                        }
                }
});

JButton btnModificar = new JButton("Modificar");
btnModificar.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent e)
        {

                try
                {
                        cn = conexion.Conectar();
                        stm =cn.createStatement();

                        String strCodigo = txtCodigo.getText();
                        String strFecha = txtFecha.getText();
                        String strGrados = txtGrados.getText();

                        String strSql = "UPDATE tbltemperatura ";
                        strSql = strSql + "SET fechtempe ='"+strFecha+"', ";
                        strSql = strSql + "gradtempe ="+strGrados+" ";
                        strSql = strSql + "WHERE codtempe= "+strCodigo;
                        stm.executeUpdate(strSql);

                        tbltempe.setModel(new DefaultTableModel(
                                        new Object[][] {
                                        },
                                        new String[] {
                                                "C\u00F3digo", "Fecha", "Grados"
                                        }
                                ));
                        scrollPane.setViewportView(tbltempe);

                        mostrarRegistros();
                }
                catch (SQLException e4)
                {
                        e4.printStackTrace();
                }
        }
});
btnModificar.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
        }
});

JButton btnSalir = new JButton("Salir");
btnSalir.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {
        }
});
btnSalir.addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent e) {
```

```
                        dispose();
                    }
                });

                JButton btnEliminar = new JButton("Eliminar");
                btnEliminar.addMouseListener(new MouseAdapter() {
                    @Override
                    public void mouseClicked(MouseEvent e)
                    {
                        try
                        {
                            cn = conexion.Conectar();
                            stm =cn.createStatement();

                            String strCodigo = txtCodigo.getText();
                            String strSql = "DELETE FROM tbltemperatura WHERE
codtempe = ";
                            strSql = strSql + strCodigo;

                            stm.executeUpdate(strSql);


                            tbltempe.setModel(new DefaultTableModel(
                                    new Object[][] {
                                    },
                                    new String[] {
                                        "C\u00F3digo", "Fecha", "Grados"
                                    }
                                ));
                            scrollPane.setViewportView(tbltempe);

                            mostrarRegistros();
                        }
                        catch (SQLException e5)
                        {
                            e5.printStackTrace();
                        }
                    }
                });
                btnEliminar.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent e) {
                    }
                });
                GroupLayout gl_contentPane = new GroupLayout(contentPane);
                gl_contentPane.setHorizontalGroup(
                    gl_contentPane.createParallelGroup(Alignment.TRAILING)
                        .addGroup(gl_contentPane.createSequentialGroup()
                            .addGap(9)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
                                .addComponent(lblFecha)
                                .addComponent(lblCodigo)
                                .addComponent(lblGrados))
                            .addGap(42)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
                                .addComponent(txtGrados,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                .addComponent(txtFecha,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                .addComponent(txtCodigo,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
```

```
                                        .addGap(18)

                .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
                                                .addComponent(btnGuardar)
                                                .addComponent(btnLimpiar)
                                                .addComponent(btnModificar))
                                        .addPreferredGap(ComponentPlacement.RELATED)

                .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
                                                .addComponent(btnEliminar)
                                                .addComponent(btnSalir))
                                        .addGap(98))
                                .addComponent(scrollPane, GroupLayout.DEFAULT_SIZE, 443,
Short.MAX_VALUE)
                        );
                        gl_contentPane.setVerticalGroup(
                                gl_contentPane.createParallelGroup(Alignment.LEADING)
                                        .addGroup(gl_contentPane.createSequentialGroup()

                .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
                                                .addGroup(gl_contentPane.createSequentialGroup()

                .addContainerGap(GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                .addGroup(gl_contentPane.createParallelGroup(Alignment.TRAILING)

                .addGroup(gl_contentPane.createSequentialGroup()
                                                                        .addComponent(lblCodigo)
                                                                        .addGap(8))

                .addGroup(gl_contentPane.createSequentialGroup()

                .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)

                .addComponent(txtCodigo, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE)

                .addComponent(btnLimpiar))

                .addPreferredGap(ComponentPlacement.RELATED)))

                .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                                                        .addComponent(txtFecha,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                                                        .addComponent(lblFecha)
                                                                        .addComponent(btnGuardar))

                .addPreferredGap(ComponentPlacement.RELATED)

                .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                                                        .addComponent(lblGrados)
                                                                        .addComponent(txtGrados,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                                                        .addComponent(btnModificar)))
                                                        .addGroup(gl_contentPane.createSequentialGroup()
                                                                .addGap(24)
                                                                .addComponent(btnEliminar)

                .addPreferredGap(ComponentPlacement.RELATED)
                                                                .addComponent(btnSalir)))
                                                .addPreferredGap(ComponentPlacement.UNRELATED)
```

```
                                        .addComponent(scrollPane, GroupLayout.DEFAULT_SIZE, 146,
Short.MAX_VALUE))
                );
                gl_contentPane.linkSize(SwingConstants.HORIZONTAL, new Component[] {btnLimpiar,
btnGuardar, btnModificar, btnSalir, btnEliminar});

                tbltempe = new JTable();
                tbltempe.addKeyListener(new KeyAdapter() {
                        @Override
                        public void keyPressed(KeyEvent e)
                        {
                        }

                        @Override
                        public void keyTyped(KeyEvent e)
                        {
                        }
                        @Override
                        public void keyReleased(KeyEvent e)
                        {
                                int posregtbltempe=tbltempe.getSelectedRow();
                                mostrarInfo(posregtbltempe);
                        }
                });
                tbltempe.addMouseListener(new MouseAdapter() {
                        @Override
                        public void mouseClicked(MouseEvent e)
                        {
                                int posregtbltempe=tbltempe.getSelectedRow();
                                mostrarInfo(posregtbltempe);
                        }
                });

                tbltempe.setModel(new DefaultTableModel(
                        new Object[][] {
                        },
                        new String[] {
                                "C\u00F3digo", "Fecha", "Grados"
                        }
                ));
                scrollPane.setViewportView(tbltempe);
                contentPane.setLayout(gl_contentPane);

                mostrarRegistros();
                if (tbltempe.getRowCount()>1)
                {
                        mostrarInfo(0);
                }
        }
}
```

Clase Simulación

```
import java.awt.BorderLayout;
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.GroupLayout;
import javax.swing.GroupLayout.Alignment;
import javax.swing.JScrollPane;
```

```
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import javax.swing.JLabel;
import javax.swing.LayoutStyle.ComponentPlacement;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import javax.swing.JOptionPane;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import java.util.Calendar;
import java.util.GregorianCalendar;

public class FrmSimulacion extends JFrame {

	private JPanel contentPane;
	private JTable tblsimdatos;
	private JTextField txtCod;
	private JTextField txtPeso;
	private JTextField txtDensi;
	private JTextField txtVoluTanq;
	private JTextField txtCTC;
	private JTextField txtPesoIni;
	private JTextField txtSPorc;
	private JTextField txtPfg;
	private JTextField txtProdAnual;
	private JTextField txtNumLote;
	private JTextField txtNumPecesIni;
	private JTextField txtMigas;
	private JTextField txtPienso;
	private JTextField txtTasaEfect;
	private JTextField txtEcotex;

	int Peso1 = 3;
	int Peso2 = 20;
	int Peso3 = 300;

	int Densi1 = 5;
	int Densi2 = 15;
	int Densi3 = 25;

	int VoluTanq1 = 60;
	int VoluTanq2 = 150;
	int VoluTanq3 = 300;

	int TasaEfect = 18;

	double CTC = 0.00228;
	int PesoIni = 1;
	int SPorc = 88;
	int Pfg = 300;
	int ProdAnual =60;

	int NumLote = 6;
	long NumPecesIni = ProdAnual*1000000/Pfg/SPorc*100/NumLote;
```

```
String Ecotex= "ecotex30";
double Migas = 0.85;
double Pienso = 0.70;

int Dias=0;
int TempMedia = 32;
int STempEfect=0;
double Peso=0.00;

int Superv = 100-2;
long Tilapia=0;
double Biomasa=0.00;

int Densidad=0;
double Volumen=0.00;
double VoluTanq=0.00;
double NumTanqTeor=0.00;
int NumTanqReal = 1;
double PesoMedio=0.00;
double TipAlim = 5.0;
double BioMed=0.00;
double PiensoKG=0.00;
String TipoPienso= "";
double PrecioPienso=0.00;

double BiomasaAnt=0.00;
double PesoAnt=0.00;

Conexion conexion = new Conexion();
Connection cn=null;
Statement stm=null;
ResultSet rs=null;

public double [] gradosmes = new double [12];

//Cargar dato de pruerba
void CargarDatos()
{
	txtCod.setText(String.valueOf(1));
	txtPeso.setText(String.valueOf(Peso1));
	txtDensi.setText(String.valueOf(Densi1));
	txtVoluTanq.setText(String.valueOf(VoluTanq1));
	txtTasaEfect.setText(String.valueOf(TasaEfect));
	txtCTC.setText(String.valueOf(CTC));
	txtPesoIni.setText(String.valueOf(PesoIni));
	txtSPorc.setText(String.valueOf(SPorc));
	txtPfg.setText(String.valueOf(Pfg));
	txtProdAnual.setText(String.valueOf(ProdAnual));
	txtNumLote.setText(String.valueOf(NumLote));
	txtNumPecesIni.setText(String.valueOf(NumPecesIni));
	txtMigas.setText(String.valueOf(Migas));
	txtEcotex.setText(String.valueOf(Ecotex));
	txtPienso.setText(String.valueOf(Pienso));
}

// Recupera temperatiras promedio mensual
void PromTempMes()
{
	try
	{
		int posGrad=0;
		cn = conexion.Conectar();
```

```
                        stm =cn.createStatement();
                        String strSql = "SELECT AVG(gradtempe) AS mediatemp FROM tbltemperatura
GROUP BY MONTH(fechtempe) ORDER BY MONTH(fechtempe) ASC";
                        rs=stm.executeQuery(strSql);

                        while(rs.next())
                        {
                                gradosmes[posGrad] = rs.getInt(1);
                                posGrad++;
                        }
                }

                catch (SQLException e)
                {
                        e.printStackTrace();
                }
                finally
                {
        try
        {
            if (rs!= null)
            {
                rs.close();
            }

            if (stm != null)
            {
                stm.close();
            }
            if (cn != null)
            {
                cn.close();
            }
        }
        catch (Exception e2)
        {
            e2.printStackTrace();
        }
    }
      }

      /**
       * Launch the application.
       */
      public static void main(String[] args) {
                EventQueue.invokeLater(new Runnable() {
                        public void run() {
                                try {
                                        FrmSimulacion frame = new FrmSimulacion();
                                        frame.setVisible(true);
                                } catch (Exception e) {
                                        e.printStackTrace();
                                }
                        }
                });
      }

      /**
       * Create the frame.
       */
      public FrmSimulacion() {
                setResizable(false);
```

```
		setAlwaysOnTop(true);
		setTitle("Simulaci\u00F3n");
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setBounds(100, 100, 1164, 593);
		contentPane = new JPanel();
		contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
		setContentPane(contentPane);

		JScrollPane scrollPane = new JScrollPane();
		scrollPane.addMouseListener(new MouseAdapter() {
			@Override
			public void mouseClicked(MouseEvent arg0) {
			}
		});

		String[] Meses = {"Enero","Febrero", "Marzo", "Abril", "Mayo","Junio","Julio","Agosto",
"Septiembre", "Octubre", "Noviembre","Diciembre"};

		JLabel lblCodigo = new JLabel("C\u00F3digo");
		JLabel lblPeso = new JLabel("Peso (g)");
		JLabel lblDensidad = new JLabel("Densidad (kg/m3)");
		JLabel lblVolumen = new JLabel("Volumen (m3)");
		JLabel lblCtc = new JLabel("CTC");
		JLabel lblPesoInicial = new JLabel("Peso Inicial (g)");
		JLabel lblS = new JLabel("S(%)");
		JLabel lblPf = new JLabel("pf (g)");
		JLabel lblProduccion = new JLabel("Producci\u00F3n Anual (tm)  ");
		JLabel lblNumeroLotes = new JLabel("N\u00FAmero Lotes");
		JLabel lblNumeroPecesInicial = new JLabel("N\u00FAmero Peces Inicial");

		txtCod = new JTextField();
		txtCod.setColumns(10);

		txtPeso = new JTextField();
		txtPeso.setColumns(10);

		txtDensi = new JTextField();
		txtDensi.setColumns(10);

		txtVoluTanq = new JTextField();
		txtVoluTanq.setColumns(10);

		txtCTC = new JTextField();
		txtCTC.setColumns(10);

		txtPesoIni = new JTextField();
		txtPesoIni.setColumns(10);

		txtSPorc = new JTextField();
		txtSPorc.setColumns(10);

		txtPfg = new JTextField();
		txtPfg.setColumns(10);

		txtProdAnual = new JTextField();
		txtProdAnual.setColumns(10);

		txtNumLote = new JTextField();
		txtNumLote.setColumns(10);

		txtNumPecesIni = new JTextField();
		txtNumPecesIni.setColumns(10);
```

```
		JLabel lblMigas = new JLabel("Migas");

		txtMigas = new JTextField();
		txtMigas.setColumns(10);

		JLabel lblPienso = new JLabel("Pienso");

		txtPienso = new JTextField();
		txtPienso.setColumns(10);



		JButton btnGenerar = new JButton("Generar");
		btnGenerar.addMouseListener(new MouseAdapter() {
			@Override
			public void mouseClicked(MouseEvent arg0)
			{
				int numCols = 19;
				Object [] fila = new Object[numCols];
				double [] PesoReg = new double[100];
				double [] BiomasaReg = new double[100];


				PromTempMes();

				try
				{
					TasaEfect = Integer.parseInt(txtTasaEfect.getText());
					PesoIni = Integer.parseInt(txtPesoIni.getText());
					CTC = Double.parseDouble(txtCTC.getText());
					SPorc = Integer.parseInt(txtSPorc.getText());
					Pfg = Integer.parseInt(txtPfg.getText());
					ProdAnual = Integer.parseInt(txtProdAnual.getText());
					NumLote = Integer.parseInt(txtNumLote.getText());
					NumPecesIni = Long.parseLong(txtNumPecesIni.getText());
					Migas = Double.parseDouble(txtMigas.getText());
					Pienso = Double.parseDouble(txtPienso.getText());

					int x=1;
					boolean continua=true;
					Calendar fecha = new GregorianCalendar();
					int anio = fecha.get(Calendar.YEAR);
					int mes = fecha.get(Calendar.MONTH);

					int PosFilx=tblsimdatos.getRowCount();
					for (int i = 0;i<PosFilx; i++)
					{
						((DefaultTableModel)
tblsimdatos.getModel()).removeRow(0);
					}

					do
					{
						int PosFil=tblsimdatos.getRowCount();

						// Generar Días
						if (mes==0 || mes==2 || mes==4 || mes==6 || mes==7 ||
mes==9 || mes==11)
							Dias = 31;
						else if(mes==1)
							{
```

```
                                        Dias = 28;
                                        if (anio%4==0) Dias=29;
                                }
                        else Dias = 30;

                        // Generar Meses
                        fila[0]=Meses[mes]+"/"+String.valueOf(anio);

                        fila[1] = Dias;

                        // Temperatura Media
                        TempMedia= (int) gradosmes[mes];
                        fila[2] =  TempMedia;

                        // S. Temperatura Efectiva
                        STempEfect =Dias*(TempMedia-TasaEfect);
                        fila[3] =STempEfect;

                        // Peso
                        if (x==1)
                        {
                                Peso =
Math.pow((Math.cbrt(PesoIni)+(STempEfect*CTC)),3);
                        }
                        else
                        {
                                PesoAnt = PesoReg[x-1];
                                Peso =
Math.pow((Math.cbrt(PesoAnt)+(STempEfect*CTC)),3);
                        }
                        fila[4] =Peso;
                        PesoReg[x]=Peso;

                        if (Peso>Pfg)
                                continua=false;

                        // Supervivencia
                        fila[5] =  Superv;

                        // Tilapia
                        Tilapia = Superv*NumPecesIni/100;
                        fila[6] = Tilapia;

                        // Biomasa
                        Biomasa = Tilapia*Peso/1000;
                        fila[7] = Biomasa;
                        BiomasaReg[x]=Biomasa;

                        // Densidad
                        if (Peso<=Peso1) Densidad = Densi1;
                        else if (Peso<=Peso2) Densidad = Densi2;
                        else Densidad = Densi3;
                        fila[8] = Densidad;

                        // Volumen
                        Volumen = Biomasa/Densidad;
                        fila[9] = Volumen;

                        // Volumen de Tanque
                        if (Densidad<=Densi1) VoluTanq= VoluTanq1;
                        else if (Densidad<=Densi2) VoluTanq= VoluTanq2;
                        else VoluTanq= VoluTanq3;
```

```
fila[10] = VoluTanq;

// Número de Tanque Teórico
NumTanqTeor = Volumen/VoluTanq;
fila[11] =          NumTanqTeor;

// Número de Tanque Real
fila[12] =          NumTanqReal;

// Peso Medio

if (x==1)
{
        PesoMedio = (Peso+PesoIni)/2;
}
else
{
        PesoAnt = PesoReg[x-1];
        PesoMedio = (Peso+PesoAnt)/2;
}
fila[13] =          PesoMedio;

// Generar Tipo de Alimentación
TipAlim = 1.2;

if (PosFil==0) TipAlim = 5;
if (PosFil==1) TipAlim = 4;
if (PosFil==2) TipAlim = 4;
if (PosFil==3) TipAlim = 3;
if (PosFil==4) TipAlim = 3;
if (PosFil==5) TipAlim = 3;
if (PosFil==6) TipAlim = 2;
if (PosFil==7) TipAlim = 2;
if (PosFil==8) TipAlim = 1.4;
if (PosFil==9) TipAlim = 1.2;
if (PosFil==10)TipAlim= 1.2;
if (PosFil==11) TipAlim = 1.2;
fila[14] =          TipAlim;

if (x==1)
{
        BioMed =
((PesoIni*NumPecesIni)/1000+Biomasa)/2;
}
else
{
        BiomasaAnt = BiomasaReg[x-1];
        BioMed = (BiomasaAnt+Biomasa)/2;
}

fila[15] =          BioMed;

PiensoKG = (Dias*TipAlim*BioMed)/100;
fila[16] =          PiensoKG;

if (Peso<8)
{
        fila[17] =          "migas";
        PrecioPienso=PiensoKG*Migas;
}
else
{
```

```
                                                fila[17] =              "pienso";
                                                PrecioPienso=PiensoKG*Pienso;
                                        }
                                        fila[18]=PrecioPienso;

                                        if (Superv>1) Superv--;
                                        if (PosFil>=0) TempMedia=25;


                                        ((DefaultTableModel)
tblsimdatos.getModel()).addRow(fila);

                                        x++;
                                        mes++;

                                        if (mes>11)
                                        {
                                                mes=0;
                                                anio++;
                                        }
                                }while(continua==true);
                        }
                        catch (Exception e)
                        {
                                JOptionPane.showMessageDialog(null, "Ingrese valores");
                        }

                }
        });
        btnGenerar.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent arg0) {
                }
        });

        JLabel lblTasaEfectiva = new JLabel("Tasa Efectiva");

        txtTasaEfect = new JTextField();
        txtTasaEfect.setColumns(10);

        JLabel lblEcotex = new JLabel("Ecotex");

        txtEcotex = new JTextField();
        txtEcotex.setColumns(10);

        JButton btnSalir = new JButton("Salir");
        btnSalir.addMouseListener(new MouseAdapter() {
                @Override
                public void mouseClicked(MouseEvent arg0) {
                        dispose();
                }
        });
        GroupLayout gl_contentPane = new GroupLayout(contentPane);
        gl_contentPane.setHorizontalGroup(
                gl_contentPane.createParallelGroup(Alignment.LEADING)
                        .addGroup(gl_contentPane.createSequentialGroup()
                                .addContainerGap()

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
                                        .addComponent(lblCodigo)
                                        .addComponent(lblPeso)
                                        .addComponent(lblDensidad)
                                        .addComponent(lblVolumen)
```

```
                                        .addComponent(lblTasaEfectiva))
                                    .addGap(18)

            .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
                                        .addComponent(txtCod,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                        .addComponent(txtPeso,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                        .addComponent(txtDensi,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                        .addComponent(txtVoluTanq,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                        .addComponent(txtTasaEfect,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
                                    .addGap(48)

            .addGroup(gl_contentPane.createParallelGroup(Alignment.TRAILING)

            .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING, false)

            .addGroup(gl_contentPane.createSequentialGroup()
                                                .addComponent(lblPf)

            .addPreferredGap(ComponentPlacement.RELATED, 88, Short.MAX_VALUE)
                                                .addComponent(txtPfg,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))

            .addGroup(gl_contentPane.createSequentialGroup()

            .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
                                                    .addComponent(lblS)
                                                    .addComponent(lblCtc)

            .addComponent(lblPesoInicial))
                                                .addGap(45)

            .addGroup(gl_contentPane.createParallelGroup(Alignment.TRAILING)

            .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)

            .addComponent(txtCTC, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE)

            .addComponent(txtPesoIni, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
                                                    .addComponent(txtSPorc,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))))
                                        .addGroup(gl_contentPane.createSequentialGroup()
                                            .addComponent(lblProduccion)

            .addPreferredGap(ComponentPlacement.UNRELATED)
                                            .addComponent(txtProdAnual,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)))
                                    .addGap(18)

            .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
                                        .addComponent(lblMigas)
                                        .addComponent(lblNumeroPecesInicial)
                                        .addComponent(lblNumeroLotes)
                                        .addComponent(lblEcotex)
                                        .addComponent(lblPienso))
                                    .addGap(10)
```

```
        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
                                                .addComponent(txtPienso,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                                .addGroup(gl_contentPane.createSequentialGroup()

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
                                                                .addComponent(txtNumLote,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)

        .addComponent(txtNumPecesIni, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE)
                                                                .addComponent(txtMigas,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
                                                                .addComponent(txtEcotex,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
                                                        .addGap(36)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
                                                                .addComponent(btnGenerar,
GroupLayout.DEFAULT_SIZE, 105, Short.MAX_VALUE)
                                                                .addComponent(btnSalir,
GroupLayout.DEFAULT_SIZE, 105, Short.MAX_VALUE))))
                                                .addGap(168))
                                        .addComponent(scrollPane, GroupLayout.DEFAULT_SIZE, 974,
Short.MAX_VALUE)
                );
                gl_contentPane.setVerticalGroup(
                        gl_contentPane.createParallelGroup(Alignment.LEADING)
                                .addGroup(gl_contentPane.createSequentialGroup()

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
                                                .addGroup(gl_contentPane.createSequentialGroup()
                                                        .addGap(22)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                                                .addComponent(lblCodigo)
                                                                .addComponent(lblCtc)
                                                                .addComponent(txtCod,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                                                .addComponent(txtCTC,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                                                .addComponent(txtNumLote,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                                                .addComponent(lblNumeroLotes))

        .addPreferredGap(ComponentPlacement.UNRELATED)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                                                .addComponent(lblPeso)
                                                                .addComponent(txtPeso,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                                                .addComponent(txtPesoIni,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)

        .addComponent(lblPesoInicial))

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
```

```
        .addComponent(txtNumPecesIni, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE)

        .addComponent(lblNumeroPecesInicial)))

        .addPreferredGap(ComponentPlacement.UNRELATED)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                                        .addComponent(lblDensidad)
                                                        .addComponent(lblS)
                                                        .addComponent(txtDensi,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                                        .addComponent(txtSPorc,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                                        .addComponent(lblEcotex)
                                                        .addComponent(txtEcotex,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)))
                                                .addGroup(gl_contentPane.createSequentialGroup()
                                                        .addGap(34)
                                                        .addComponent(btnGenerar)

        .addPreferredGap(ComponentPlacement.UNRELATED)
                                                        .addComponent(btnSalir)))
                                                .addPreferredGap(ComponentPlacement.UNRELATED)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                                        .addComponent(lblVolumen)
                                                        .addComponent(lblPf)
                                                        .addComponent(txtVoluTanq,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                                        .addComponent(txtPfg,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                                        .addComponent(lblMigas))
                                                .addComponent(txtMigas,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
                                        .addPreferredGap(ComponentPlacement.RELATED)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.TRAILING)
                                                .addGroup(gl_contentPane.createSequentialGroup()

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                                        .addComponent(lblTasaEfectiva)
                                                        .addComponent(txtTasaEfect,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
                                                        .addGap(11))

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                                        .addComponent(txtProdAnual,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                                                        .addComponent(lblProduccion)
                                                        .addComponent(lblPienso)
                                                        .addComponent(txtPienso,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)))
                                                .addPreferredGap(ComponentPlacement.RELATED)
                                                .addComponent(scrollPane, GroupLayout.DEFAULT_SIZE, 376,
Short.MAX_VALUE))
                        );

                        tblsimdatos = new JTable();
```

```
                tblsimdatos.addMouseListener(new MouseAdapter() {
                        @Override
                        public void mouseClicked(MouseEvent e) {
                        }
                });
                tblsimdatos.setModel(new DefaultTableModel(
                        new Object[][] {},
                        new String[] {
                                "Meses", "Días", "Temp. Media (ºC)", "Sum. Temp. Efect.(ºC)",
                                "Peso (g)", "Supervivencia (%)", "Tilapia", "Biomasa (Kg)",
                                "Densidad (Kg/m3)", "Volumen (m3)", "Volumen Tanque (m3)",
"Núm. Tanque Teóricos",
                                "Núm. Tanque Real", "Peso medio", "Tipo alimentación", "BioMed
(Kg)",
                                "Pienso (Kg)", "Tipo Pienso", "Precio Pienso"
                        })
                {
                        Class[] columnTypes = new Class[] {
                                String.class, Integer.class, Integer.class, Integer.class,
                                Double.class, Integer.class, Long.class, Double.class,
                                Integer.class, Integer.class, Double.class, Double.class,
                                Integer.class, Double.class, Double.class, Double.class,
                                Double.class, String.class, Double.class
                        };
                        public Class getColumnClass(int columnIndex)
                        {
                                return columnTypes[columnIndex];
                        }
                });
                scrollPane.setViewportView(tblsimdatos);
                contentPane.setLayout(gl_contentPane);

                CargarDatos();
        }
}
```