

# Prueba de Caja Blanca

---

“Registro de aplicación de vacunas contra el COVID-19  
en zonas rurales del cantón Quito”

Versión 1.1

**Integrantes:**

Aguirre Ángel  
Castillo Joel  
Castro Bryan  
Oviedo Steven

**Fecha 2021-08-27**

## Ingreso al sistema, Registro Doctores, Registro de dosis, Registro Vacunas.

### CÓDIGO FUENTE

```
public Login() {
    initComponents();
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
Generated Code

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    System.exit(0);
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    if (!verificar()) {
        try {
            // TODO add your handling code here:
            EmpleadoDAO dao = new EmpleadoDAO();
            EmpleadoDTO dto = new EmpleadoDTO();

            dto = dao.buscarCedula(jTextField1.getText());
            String pass = jPasswordField1.getText();
            if (dto != null) {
                if (dto.getPassword().equals(pass)) {
                    new Principal().setVisible(true);
                    this.setVisible(false);
                } else {
                    JOptionPane.showMessageDialog(null, "Usuario/contraseña incorrecto");
                }
            } else {
                JOptionPane.showMessageDialog(null, "Usuario/contraseña incorrecto");
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    } else {
        JOptionPane.showMessageDialog(null, "Te falta completar algún campo requerido. Por favor verifica.");
    }
}

private boolean verificar() {
    return jTextField1.getText().equals("") || jPasswordField1.getText().equals(" ");
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    //Crear cuenta nueva
    new Registro(this, true).setVisible(true);
}

private void jPasswordField1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

public Image getIconImage() {
    Image retValue = Toolkit.getDefaultToolkit().getImage(ClassLoader.getResource("imagenes/FondoAzul.png"));
    return retValue;
}
```

```

public class Registro extends javax.swing.JDialog {

    /**
     * Creates new form Registro
     */
    public Registro(java.awt.Frame parent, boolean modal) {
        super(parent, modal);
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        //Registrar
        if (!verificar()) {
            if (verificarCorreo()) {
                if (isEcuadorianDocumentValid(jTextField3.getText())) {
                    try {
                        EmpleadoDAO dao = new EmpleadoDAO();
                        EmpleadoDTO dto = new EmpleadoDTO();

                        dto.setNombre(jTextField1.getText());
                        dto.setApellido(jTextField2.getText());
                        dto.setCedula(jTextField3.getText());
                        dto.setCorreo(jTextField4.getText());
                        dto.setCorreo(jTextField4.getText());
                        dto.setTelefono(jTextField5.getText());
                        dto.setPassword(jTextField6.getText());
                        dao.append(dto);
                        limpiar();
                        JOptionPane.showMessageDialog(null, "Cliente Agregado con Exito");
                    } catch (SQLException ex) {
                        JOptionPane.showMessageDialog(null, "Esta cedula ya existe");
                        ex.printStackTrace();
                    } catch (Exception ex) {
                        JOptionPane.showMessageDialog(null, "Datos Incorrectos");
                    }
                } else {
                    JOptionPane.showMessageDialog(null, "La cedula no es valida");
                }
            } else {
                JOptionPane.showMessageDialog(null, "El correo no es valido");
            }
        } else {
            JOptionPane.showMessageDialog(null, "Te falta completar algún campo requerido. Por favor verifica.");
        }
    }

    private void jTextField3ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void limpiar() {
        jTextField1.setText("");
        jTextField2.setText("");
        jTextField3.setText("");
    }
}

```

```

        jTextField3.setText("");
        jTextField4.setText("");
        jTextField5.setText("");
        jTextField6.setText("");
    }

    private boolean verificar() {
        return jTextField1.getText().equals("") || jTextField2.getText().equals("") || jTextField3.getText().equals("")
            || jTextField4.getText().equals("") || jTextField5.getText().equals("") || jTextField6.getText().eq
    }

    private boolean verificarCorreo() {
        return jTextField4.getText().contains("@");
    }

    private boolean isEcuadorianDocumentValid(String document) {
        byte sum = 0;
        try {
            if (document.trim().length() != 10) {
                return false;
            }
            String[] data = document.split("");
            byte verifier = Byte.parseByte(data[0] + data[1]);
            if (verifier < 1 || verifier > 24) {
                return false;
            }
            byte[] digits = new byte[data.length];
            for (byte i = 0; i < digits.length; i++) {
                digits[i] = Byte.parseByte(data[i]);
            }
            if (digits[2] > 6) {
                return false;
            }
            for (byte i = 0; i < digits.length - 1; i++) {
                if (i % 2 == 0) {
                    verifier = (byte) (digits[i] * 2);
                    if (verifier > 9) {
                        verifier = (byte) (verifier - 9);
                    }
                } else {
                    verifier = (byte) (digits[i] * 1);
                }
                sum = (byte) (sum + verifier);
            }
            if ((sum - (sum % 10) + 10 - sum) == digits[9]) {
                return true;
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return false;
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)
    }

```

```

public class RegistroDosis extends javax.swing.JInternalFrame {

    DosisDAO dao = new DosisDAO();
    DosisDTO dto = new DosisDTO();

    public RegistroDosis() {
        initComponents();
        actualizarTabla();
        actualizarComboBox();
        java.util.Date utilDate = new java.util.Date();
        java.sql.Date sqlDate = new java.sql.Date(utilDate.getTime());
        DateFormat date = new SimpleDateFormat("yyyy-MM-dd");
        jDateChooser1.setDate(sqlDate);
        actualizarTabla2();
    }

    private void actualizarTabla() {
        try {
            List<DosisDTO> lista = dao.readALL();

            String matriz[][] = new String[lista.size()][9];

            for (int i = 0; i < lista.size(); i++) {
                matriz[i][0] = lista.get(i).getId() + "";
                matriz[i][1] = lista.get(i).getCedula();
                matriz[i][2] = lista.get(i).getNombrePersona();
                matriz[i][3] = lista.get(i).getEdad() + "";
                matriz[i][4] = lista.get(i).getVacuna();
                matriz[i][5] = lista.get(i).getNumDosis() + "";
                matriz[i][6] = lista.get(i).getCorreo();
                matriz[i][7] = lista.get(i).getTelefono();
                matriz[i][8] = lista.get(i).getFecha();
            }

            jTable1.setModel(new javax.swing.table.DefaultTableModel(
                matriz,
                new String[]{
                    "ID", "Cedula", "Nombre", "Edad", "Tipo vacuna", "N dosis suministrada", "correo", "Telefon
                });
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }

    private void actualizarTabla2() {
        try {
            List<DosisDTO> lista = dao.readALL();

            String matriz[][] = new String[lista.size()][9];

            for (int i = 0; i < lista.size(); i++) {
                Date fechaactual = new Date(System.currentTimeMillis());
                SimpleDateFormat formato = new SimpleDateFormat("yyyy-MM-dd");
                Date fecha = formato.parse(lista.get(i).getFecha());
                int millisecondsByDay = 86400000;
                int dias = (int) ((fechaactual.getTime() - fecha.getTime()) / millisecondsByDay);
                if (dias > 25) {
                    matriz[i][0] = lista.get(i).getId() + "";
                    matriz[i][1] = lista.get(i).getCedula();
                    matriz[i][2] = lista.get(i).getNombrePersona();
                    matriz[i][3] = lista.get(i).getEdad() + "";
                }
            }
        }
    }
}

```

```

        matriz[i][4] = lista.get(i).getVacuna();
        matriz[i][5] = lista.get(i).getNumDosis() + "";
        matriz[i][6] = lista.get(i).getCorreo();
        matriz[i][7] = lista.get(i).getTelefono();
        matriz[i][8] = lista.get(i).getFecha();
    }
}

jTable2.setModel(new javax.swing.table.DefaultTableModel(
    matriz,
    new String[]{
        "ID", "Cedula", "Nombre", "Edad", "Tipo vacuna", "N dosis suministrada", "correo", "Telefon
    });
} catch (Exception ex) {
    ex.printStackTrace();
}
}

private void actualizarComboBox() {
    try {
        VacunaDAO dao = new VacunaDAO();
        List<VacunaDTO> lista = dao.readALL();
        for (int i = 0; i < lista.size(); i++) {
            vacunaBox.addItem(lista.get(i).getNombre());
        }
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    //Registrar
    if (!verificar()) {
        if (verificarCorreo()) {
            if (isEcuadorianDocumentValid(cedulaText.getText())) {
                try {
                    DateFormat date = new SimpleDateFormat("yyyy-MM-dd");
                    String fecha = date.format(jDateChooser1.getDate());
                    dto.setNombrePersona(nombreText.getText());
                    dto.setEdad(Integer.parseInt(edadText.getText()));
                    dto.setCedula(cedulaText.getText());
                    dto.setCorreo(correoText.getText());
                    dto.setTelefono(telefonoText.getText());
                    dto.setVacuna(vacunaBox.getSelectedItem().toString());
                    dto.setNumDosis(1);
                    dto.setFecha(fecha);
                    dao.append(dto);
                    actualizarTabla();
                    actualizarTabla2();
                    limpiar();
                    JOptionPane.showMessageDialog(null, "Registro exitoso");
                } catch (SQLException ex) {
                    JOptionPane.showMessageDialog(null, "Ha ocurrido un error inesperado");
                    ex.printStackTrace();
                }
            } else {
                JOptionPane.showMessageDialog(null, "El número cédula no es valida");
            }
        } else {
            JOptionPane.showMessageDialog(null, "El correo no es valido");
        }
    }
}

```

```

    } else {
        JOptionPane.showMessageDialog(null, "Te falta completar algún campo requerido. Por favor verifica.");
    }
}

private boolean verificar() {
    return nombreText.getText().equals("") || edadText.getText().equals("") || cedulaText.getText().equals("")
        || correoText.getText().equals("") || telefonoText.getText().equals("") || vacunaBox.getSelectedIndex() == -1;
}

private boolean verificarCorreo() {
    return correoText.getText().contains("@");
}

private boolean isEcuadorianDocumentValid(String document) {
    byte sum = 0;
    try {
        if (document.trim().length() != 10) {
            return false;
        }
        String[] data = document.split("");
        byte verifier = Byte.parseByte(data[0] + data[1]);
        if (verifier < 1 || verifier > 24) {
            return false;
        }
        byte[] digits = new byte[data.length];
        for (byte i = 0; i < digits.length; i++) {
            digits[i] = Byte.parseByte(data[i]);
        }
        if (digits[2] > 6) {
            return false;
        }
        for (byte i = 0; i < digits.length - 1; i++) {
            if (i % 2 == 0) {
                verifier = (byte) (digits[i] * 2);
                if (verifier > 9) {
                    verifier = (byte) (verifier - 9);
                }
            } else {
                verifier = (byte) (digits[i] * 1);
            }
            sum = (byte) (sum + verifier);
        }
        if ((sum - (sum % 10) + 10 - sum) == digits[9]) {
            return true;
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return false;
}

private void limpiar()
{
    nombreText.setText("");
    edadText.setText("");
    cedulaText.setText("");
    telefonoText.setText("");
    correoText.setText("");
    IDLabel.setText("");
}

```

```

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    //Actualizar
    if (!IDLabel.getText().equals("")) {
        if (verificarCorreo()) {
            if (isEcuadorianDocumentValid(cedulaText.getText())) {
                try {
                    DateFormat date = new SimpleDateFormat("yyyy-MM-dd");
                    String fecha = date.format(jDateChooser1.getDate());
                    dto.setNombrePersona(nombreText.getText());
                    dto.setEdad(Integer.parseInt(edadText.getText()));
                    dto.setCedula(cedulaText.getText());
                    dto.setCorreo(correoText.getText());
                    dto.setTelefono(telefonoText.getText());
                    dto.setVacuna(vacunaBox.getSelectedItem().toString());
                    dto.setNumDosis(1);
                    dto.setFecha(fecha);
                    dto.setId(Integer.parseInt(IDLabel.getText()));
                    dao.update(dto);
                    actualizarTabla();
                    actualizarTabla2();
                    limpiar();
                    JOptionPane.showMessageDialog(null, "Actualizacion exitosa");
                } catch (SQLException ex) {
                    JOptionPane.showMessageDialog(null, "Ha ocurrido un error inesperado");
                    ex.printStackTrace();
                }
            } else {
                JOptionPane.showMessageDialog(null, "El número de cédula no es valido");
            }
        } else {
            JOptionPane.showMessageDialog(null, "El correo no es valido");
        }
    } else {
        JOptionPane.showMessageDialog(null, "Te falta completar algún campo requerido. Por favor verifica.");
    }
}

```

```

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
    //mouse clicked
    int fila = jTable1.getSelectedRow();
    String id = jTable1.getValueAt(fila, 0).toString();
    IDLabel.setText(id);
    cedulaText.setText(jTable1.getValueAt(fila, 1).toString()); //CEDULA
    nombreText.setText(jTable1.getValueAt(fila, 2).toString()); //NOMBRE
    edadText.setText(jTable1.getValueAt(fila, 3).toString()); //EDAD
    //correoText.setText(jTable1.getValueAt(fila, 4).toString()); //TIPO VACUNA
    //telefonoText.setText(jTable1.getValueAt(fila, 5).toString()); //NDOSIS
    correoText.setText(jTable1.getValueAt(fila, 6).toString()); //CORREO
    telefonoText.setText(jTable1.getValueAt(fila, 7).toString()); //TELEFONO
}

```

```

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    //Eliminar
    try {
        int fila = jTable1.getSelectedRow();
        String id = jTable1.getValueAt(fila, 0).toString();
        dao.delete(Integer.parseInt(id));
        JOptionPane.showMessageDialog(null, "Registro Eliminado");
        actualizarTabla();
        actualizarTabla2();
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null, "Error inesperado");
    }
}

```

```

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    // Eliminar Todo
    try {
        dao.deleteAll();
        actualizarTabla();
        actualizarTabla2();
        JOptionPane.showMessageDialog(null, "Todos los registros se han eliminado");
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null, "Error inesperado");
    }
}

```



```

public class RegistroVacuna extends javax.swing.JInternalFrame {

    VacunaDAO dao = new VacunaDAO();
    VacunaDTO dto = new VacunaDTO();

    public RegistroVacuna() {
        initComponents();
        actualizarTabla();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        if (!verificar()) {
            try {
                //Registrar vacuna
                dto.setNombre(jTextField1.getText());
                dto.setNumdosis(Integer.parseInt(jTextField2.getText()));
                dao.append(dto);
                actualizarTabla();
                limpiar();
                JOptionPane.showMessageDialog(null, "Registro exitoso");
            } catch (SQLException ex) {
                JOptionPane.showMessageDialog(null, "Ha ocurrido un error inesperado");
                ex.printStackTrace();
            }
        } else {
            JOptionPane.showMessageDialog(null, "Te falta completar algún campo requerido. Por favor verifica.");
        }
    }

    private boolean verificar() {
        return jTextField1.getText().equals("") || jTextField2.getText().equals("");
    }

    private void limpiar() {
        jTextField1.setText("");
        jTextField2.setText("");
    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        // Eliminar
        try {
            int fila = jTable1.getSelectedRow();
            String ID = jTable1.getValueAt(fila, 0).toString();
            dao.delete(Integer.parseInt(ID));
            JOptionPane.showMessageDialog(null, "Registro Eliminado");
            actualizarTabla();
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(null, "Error inesperado");
        }
    }

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        // Eliminar Todo
        try {
            dao.deleteAll();
        }
    }
}

```

```
        JOptionPane.showMessageDialog(null, "Registro Eliminado");
        actualizarTabla();
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null, "Error inesperado");
    }
}
```

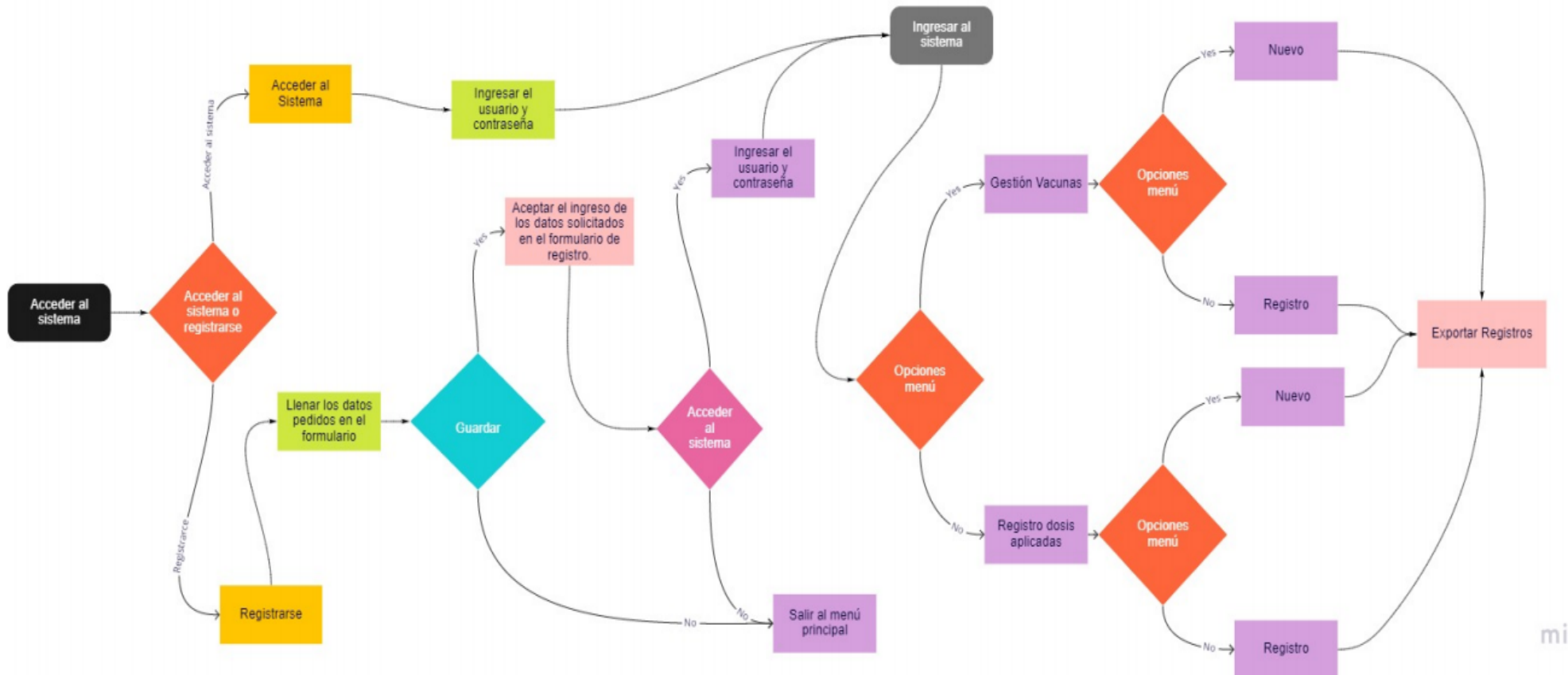
```
private void actualizarTabla() {
    try {
        List<VacunaDTO> lista = dao.readALL();

        String matriz[][] = new String[lista.size()][3];

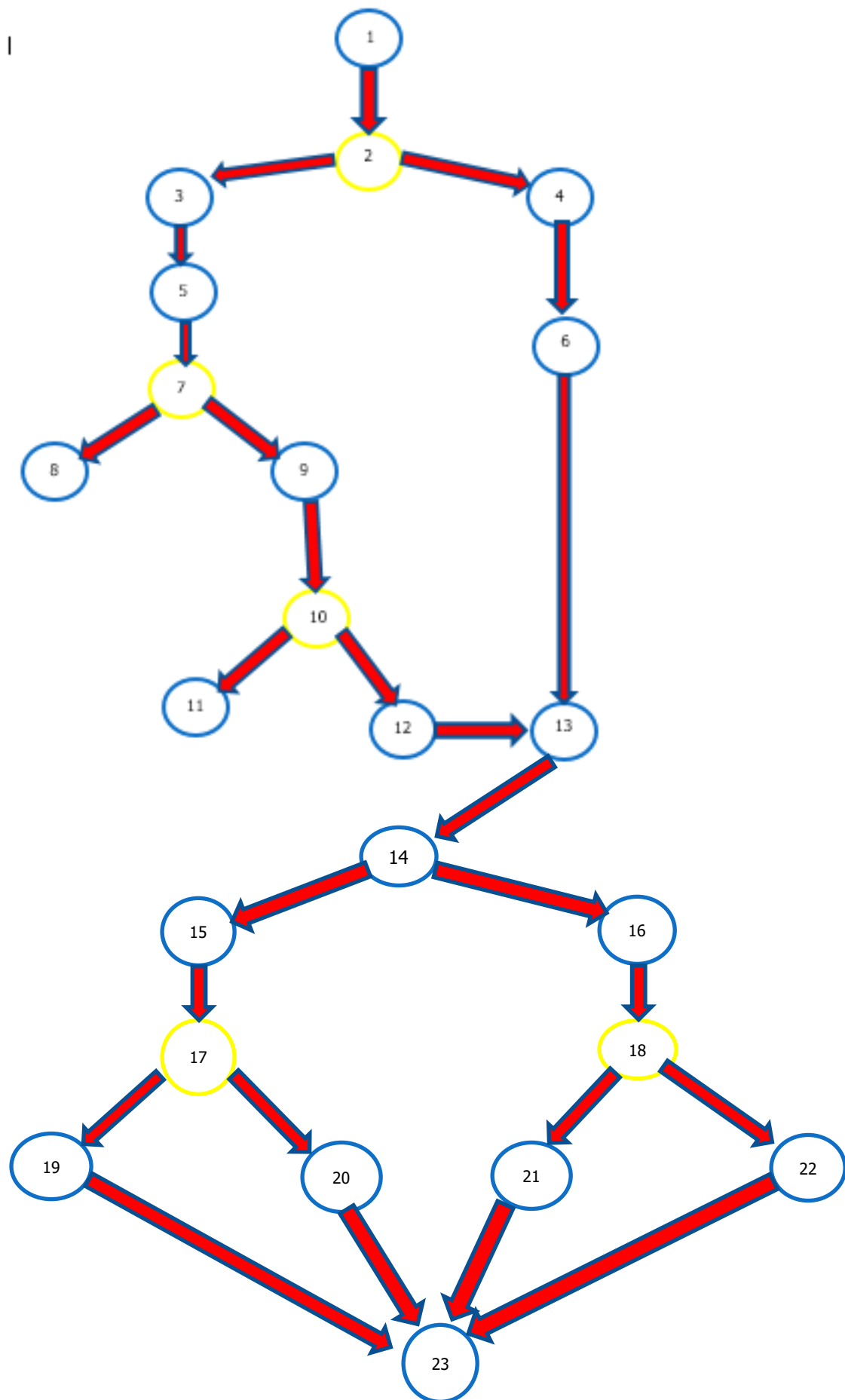
        for (int i = 0; i < lista.size(); i++) {
            matriz[i][0] = lista.get(i).getId() + "";
            matriz[i][1] = lista.get(i).getNombre();
            matriz[i][2] = lista.get(i).getNumdosis() + "";
        }

        jTable1.setModel(new javax.swing.table.DefaultTableModel(
            matriz,
            new String[]{
                "ID", "Nombre", "Cantidad de dosis"
            }
        ));
    } catch (Exception ex) {
    }
}
```

## Diagrama de Flujo



# GRAFO



## **RUTAS**

**R1:** 1,2,3,5,7,8

**R2:** 1,2,3,5,7,9,10,11

**R3:** 1,2,3,5,7,9,10,12,13,14,15,17,19,23

**R4:** 1,2,3,5,7,9,10,12,13,14,15,17,20,23

**R5:** 1,2,3,5,7,9,10,12,13,14,16,18,21,23

**R6:** 1,2,3,5,7,9,10,12,13,14,16,18,22,23

**R7:** 1,2,4,6,13,14,15,17,19,23

**R8:** 1,2,4,6,13,14,15,17,20,23

**R9:** 1,2,4,6,13,14,16,18,21,23

**R10:** 1,2,4,6,13,14,16,18,22,23

## **COMPLEJIDAD CICLOMÁTICA**

**P:** Número de nodos predicado

**A:** Número de aristas

**N:** Número de nodos

Se puede calcular de la siguiente forma:

- $V(G) = A - N + 2 = 27 - 23 + 2 = \mathbf{6}$
- $V(G) = P + 1 = \mathbf{6}$