



*Universidad Nacional de La Matanza*

Florencio Varela 1903 - San Justo - Buenos Aires - Argentina

## **Departamento de Ingeniería e Investigaciones Tecnológicas**

# ***ANÁLISIS DE SOFTWARE***

### ***Documentación Técnica***

#### **ALUMNOS:**

Anfuso Diego  
Valdéz Ezequiel  
Salías Analía  
Illobre Gonzalo

#### **PROFESORES:**

Lic. Marcelo Vinjoy  
Ing. R. Landaburu  
Lic. Enzo Delben  
Lic. Gustavo Agustín

**GRUPO:** 14

**COMISIÓN:** 5900

**TURNO:** viernes - Noche

**AULA:** 324

# ÍNDICE

Descripción Funcional .....	3
Código del Programa .....	3
DatosPacienteW.java .....	3
IngresoPacientesW.java .....	13
Login.java .....	20
MainW.java .....	25
ManualW.java .....	31
PacientesXMedW.java .....	34
SituacionPacienteW.java .....	45
bdatao.java .....	53
DatosMedicoW.java .....	54
InformesW.java .....	64
EnfermedadesMedW.java .....	71
EncryptHelper.java .....	81
Modelo de Calidad y Algoritmo .....	82
Características y Subcaracterísticas Utilizadas .....	82
Algoritmo de Calidad .....	83
Criterios de Evaluación de Métricas .....	84
Métodos de Caja Blanca .....	88
Método a Analizar .....	88
Reporte de la herramienta de Testing .....	89
Método de McCabe .....	89
Métodos de caja negra .....	91
Cubierta de CD .....	93
Pruebas de Sistema .....	95
Pruebas de recuperación .....	95
Pruebas de seguridad .....	95
Pruebas de resistencia .....	96
Pruebas de rendimiento .....	97
Encuesta de satisfacción de usuario .....	97
Funcionalidad .....	97
Eficiencia .....	97
Fiabilidad .....	97
Mantenibilidad .....	98

Usabilidad.....	98
Portabilidad.....	98
Algoritmo para obtener el puntaje final .....	98
Resultados de la encuesta.....	99
Perfil de los encuestados .....	99
Respuestas obtenidas .....	99
Calificación Obtenida .....	100
Gráfico Representativo .....	101
Resultados de Evaluación de Métricas.....	102
Puntaje obtenido mediante el Algoritmo de Calidad.....	105
Comparación de Encuesta de usuario con Modelo de calidad .....	106

## DESCRIPCIÓN FUNCIONAL

El sistema de Control de Pacientes permite guardar información sobre los pacientes y los médicos que los atienden en el centro médico “Los Laureles”, pudiendo de esta manera generar reportes acerca de las enfermedades que tratan los médicos como también saber qué pacientes se atienden con cada médico.

## CÓDIGO DEL PROGRAMA

### DATOSPACIENTEW.JAVA

```
package controlpac;

import java.awt.event.KeyEvent;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.EOFException;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;

/**
 *
 * @author Administrador
 */
public class DatosPacienteW extends javax.swing.JFrame {

    String codPac="";
    String nomPac="";
    /**
     * Creates new form DatosPacienteW
     */
    public DatosPacienteW() { //CONSTRUCTOR 1
        initComponents();
        setIconImage(new ImageIcon(getClass().getResource("/imagenes/iconoCM.png")).getImage()); //Agrego el
        icono a la ventana
        setSize(400,400); //le doy tamaño
        setResizable(false); // Anula poder maximizar la ventana.
        setLocationRelativeTo(null); // La ventana en el centro.
        setTitle("Ingreso de Datos"); // Le da un titulo a la ventana.
    }

    public DatosPacienteW(String codigo) { //CONSTRUCTOR para crear un paciente desde otras clases.
        this.codPac=codigo; // Guardo el dato pasado.
    }

    public String getCodigoPac(){
        // Obtenemos el codigo del paciente ingresado.
    }
}
```

```

        this.codPac=codigoPac.getText();
        return codPac;
    }

    public String getNombrePac(){
        // Obtenemos el nombre del paciente ingresado.
        this.nomPac=nombrePac.getText();
        return nomPac;
    }

    private static boolean soloLetras(String cadena){
        if(cadena.matches("[áéíóúña-zÑÁÉÍÓÚA-Z][áéíóúña-zÁÉÍÓÚÑA-z ]*")) // Se valida en esta función que no puede
        ser vacía la cadena.
            return true; //Son letras.
        else
            return false; //No son letras.
    }

    private static boolean alfanumerico(String cadena){
        if(cadena.matches("[0-9a-zA-z]+")) // Tampoco puede ser vacía una cadena.
            return true; // Es alfanimerico
        else
            return false; //No es alfanumerico
    }

    public int validarDatosPac(){
        String mensaje="";
        int ret=0; // comienza siendo valido.
        if(!alfanumerico(codPac)){ // El codigo del paciente debe ser alfanumerico sin espacios.
            mensaje="El código no es válido o no se ha ingresado. "; //Se guarda mensaje de aviso.
            ret++; //Ret se incrementa porque ya no es valido.
        }
        if(!soloLetras(nomPac)){ // El nombre del paciente sólo debe contener letras.
            mensaje= mensaje + "El nombre del paciente posee caracteres incorrectos o no se ha ingresado."; //Se
            guarda mensaje de aviso.
            ret++; // Ret deja de ser valido.
        }
        if(ret!=0)
            JOptionPane.showMessageDialog(null, mensaje);
        return ret; // Correctos o Incorrectos.
    }

    public boolean ExistePac(){
        boolean found = false; // Creo una variable boolena que será la que idique si existe o no el paciente.
        try {
            DataInputStream datopac = null; // Luego abro el archivo de pacientes para recorrerlo.
            datopac = new DataInputStream(new FileInputStream(System.getProperty("user.home") +
            "\\datopac.txt")); // abre el archivo de pacientes para lectura
            int sw1 = 1; // Bandera para recorrer el archivo hasta su fin.
            while (sw1 != 0) { // recorre los registros
                try {
                    String codpa = EncryptHelper.Desencriptar(datopac.readUTF()); // lee el registro
                    datopac.readUTF();
                }
            }
        }
    }

```

```

        if (codpa.equals(codPac)){
            /*compara el codigo digitado con el codigo del paciente de la
            tabla "datopac" */
            found = true; //ENCONTRADO
        }
    } catch (EOFException e) {
        sw1 = 0;
    }
}
} catch (Exception e) {
    //JOptionPane.showMessageDialog(null, "Error al verificar la existencia del paciente"); // Mensaje
    inesperado de error.
}
return found; // Se devuelve true or false.
}

public void grabarDatosPac(){
    // Esta función se encarga de grabar los datos en el archivo de pacientes.
    try{
        DataOutputStream datopac = null; //Para luego abrir el archivo de pacientes para escritura con append
        habilitado.
        datopac = new DataOutputStream(new FileOutputStream(System.getProperty("user.home") +
        "\\datopac.txt", true)); //Se abre el archivo.
        // graba los datos en el archivo
        datopac.writeUTF(EncryptHelper.Encriptar(codPac)); //Se graba el codigo de paciente.
        datopac.writeUTF(EncryptHelper.Encriptar(nomPac)); // Se graba el nombre de paciente.
        JOptionPane.showMessageDialog(null, "Datos guardados de forma exitosa.");
    } catch (Exception ioe){
        JOptionPane.showMessageDialog(null, "Error al intentar abrir el archivo de Pacientes");
    }
}

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

    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    codigoPac = new javax.swing.JTextField();
    nombrePac = new javax.swing.JTextField();
    ingresarBtn = new javax.swing.JButton();
    anterior = new javax.swing.JButton();
    borrarC = new javax.swing.JButton();
    jLabel4 = new javax.swing.JLabel();
    jMenuBar3 = new javax.swing.JMenuBar();
    jMenu4 = new javax.swing.JMenu();
    jMenuItem4 = new javax.swing.JMenuItem();

```

```

jMenu5 = new javax.swing.JMenu();
mINI = new javax.swing.JMenuItem();
mIngDat = new javax.swing.JMenuItem();
mInf = new javax.swing.JMenuItem();
opciones = new javax.swing.JMenu();
Salir = new javax.swing.JMenuItem();
cerrarSesion = new javax.swing.JMenuItem();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
getContentPane().setLayout(null);

jLabel1.setFont(new java.awt.Font("Cambria", 0, 18)); // NOI18N
jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel1.setText("DATOS DEL PACIENTE");
jLabel1.setBorder(javax.swing.BorderFactory.createEtchedBorder());
getContentPane().add(jLabel1);
jLabel1.setBounds(10, 11, 380, 26);

jLabel2.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jLabel2.setText("Código:");
getContentPane().add(jLabel2);
jLabel2.setBounds(37, 94, 48, 17);

jLabel3.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jLabel3.setText("Nombre:");
getContentPane().add(jLabel3);
jLabel3.setBounds(31, 134, 54, 17);

codigoPac.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
codigoPac.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        codigoPacActionPerformed(evt);
    }
});
codigoPac.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        codigoPacKeyTyped(evt);
    }
});
getContentPane().add(codigoPac);
codigoPac.setBounds(103, 93, 93, 21);

nombrePac.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
nombrePac.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        nombrePacKeyTyped(evt);
    }
});
getContentPane().add(nombrePac);
nombrePac.setBounds(103, 133, 210, 21);

ingresarBtn.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
ingresarBtn.setText("Ingresar");

```

```

ingresarBtn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ingresarBtnActionPerformed(evt);
    }
});
ingresarBtn.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        ingresarBtnKeyTyped(evt);
    }
});
getContentPane().add(ingresarBtn);
ingresarBtn.setBounds(275, 172, 90, 33);

anterior.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
anterior.setText("Anterior");
anterior.setMaximumSize(new java.awt.Dimension(77, 33));
anterior.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        anteriorActionPerformed(evt);
    }
});
anterior.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        anteriorKeyTyped(evt);
    }
});
getContentPane().add(anterior);
anterior.setBounds(31, 251, 100, 33);

borrarC.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
borrarC.setText("Borrar Campos");
borrarC.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        borrarCActionPerformed(evt);
    }
});
borrarC.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        borrarCKeyTyped(evt);
    }
});
getContentPane().add(borrarC);
borrarC.setBounds(130, 172, 140, 33);

jLabel4.setIcon(new javax.swing.ImageIcon(getClass().getResource("/imagenes/fondo.jpg"))); // NOI18N
getContentPane().add(jLabel4);
jLabel4.setBounds(0, -10, 530, 390);

jMenu4.setText("Ayuda");
jMenu4.setTooltipText("");

jMenuItem4.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_A,
java.awt.event.InputEvent.CTRL_MASK));

```



```

jMenuItem4.setText("Manual");
jMenuItem4.setToolTipText("Vista de guía rápida de uso.");
jMenuItem4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem4ActionPerformed(evt);
    }
});
jMenu4.add(jMenuItem4);

jMenuBar3.add(jMenu4);

jMenu5.setText("Ir a");

mINI.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_H,
java.awt.event.InputEvent.CTRL_MASK));
mINI.setText("Menu Inicio");
mINI.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        mINIActionPerformed(evt);
    }
});
jMenu5.add(mINI);

mIngDat.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_D,
java.awt.event.InputEvent.CTRL_MASK));
mIngDat.setText("Ingreso de Datos");
mIngDat.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        mIngDatActionPerformed(evt);
    }
});
jMenu5.add(mIngDat);

mInf.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_I,
java.awt.event.InputEvent.CTRL_MASK));
mInf.setText("Informes");
mInf.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        mInfActionPerformed(evt);
    }
});
jMenu5.add(mInf);

jMenuBar3.add(jMenu5);

opciones.setText("Opciones");

Salir.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_S,
java.awt.event.InputEvent.CTRL_MASK));
Salir.setText("Salir del Programa");
Salir.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        SalirActionPerformed(evt);
    }
});

```

```

    }
});
opciones.add(Salir);

cerrarSesion.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_C,
java.awt.event.InputEvent.CTRL_MASK));
cerrarSesion.setText("Cerrar Sesión");
cerrarSesion.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cerrarSesionActionPerformed(evt);
    }
});
opciones.add(cerrarSesion);

jMenuBar3.add(opciones);

setJMenuBar(jMenuBar3);

pack();
} // </editor-fold> // GEN-END: initComponents

private void anteriorActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_anteriorActionPerformed
    // TODO add your handling code here:
    IngresoPacientesW menu=new IngresoPacientesW();
    menu.setVisible(true); // Vuelve el menu de ingreso de datos
    dispose(); // Desaparece el ingreso de datos de paciente.
} // GEN-LAST:event_anteriorActionPerformed

private void ingresarBtnKeyTyped(java.awt.event.KeyEvent evt) { // GEN-FIRST:event_ingresarBtnKeyTyped
    // TODO add your handling code here:
    // Creamos un evento para que cuando se presione la tecla Enter sea como presionar el botón.
    // *****
    char cTeclaPresionada=evt.getKeyChar();
    // da click al detectar la tecla ENTER.
    if(cTeclaPresionada==KeyEvent.VK_ENTER){
        //Ejecuta el botón (dar click)
        ingresarBtn.doClick();
    } // fin del if.
} // GEN-LAST:event_ingresarBtnKeyTyped

private void anteriorKeyTyped(java.awt.event.KeyEvent evt) { // GEN-FIRST:event_anteriorKeyTyped
    // TODO add your handling code here:
    // Creamos un evento para que cuando se presione la tecla Enter sea como presionar el botón.
    // *****
    char cTeclaPresionada=evt.getKeyChar();
    // da click al detectar la tecla ENTER.
    if(cTeclaPresionada==KeyEvent.VK_ENTER){
        //Ejecuta el botón (dar click)
        anterior.doClick();
    } // fin del if.
} // GEN-LAST:event_anteriorKeyTyped

```

```

private void ingresarBtnActionPerformed(java.awt.event.ActionEvent evt) {GEN-
FIRST:event_ingresarBtnActionPerformed
    // TODO add your handling code here:
    getCodigoPac(); //Obtenemos el codigo del paciente ingresado.
    getNombrePac(); // Obtenemos el nombre del paciente ingresado.
    if(validarDatosPac()!=0){ // Si los datos estan bien ingresados, verifico la existencia del paciente.
        if(!ExistePac()) // Si el paciente no está registrado en el archivo entonces lo agrego.
            grabarDatosPac(); //Se guardan los datos ingresados en el archivo.
        else
            JOptionPane.showMessageDialog(null, "El código de paciente ya existe."); //Se informa que el paciente
ya existe.
    }
}
} //GEN-LAST:event_ingresarBtnActionPerformed

private void codigoPacActionPerformed(java.awt.event.ActionEvent evt) {GEN-
FIRST:event_codigoPacActionPerformed
    // TODO add your handling code here:
    // Creamos un evento para que cuando se presione la tecla Enter sea como presionar el botón Ingresar.
    //*****

} //GEN-LAST:event_codigoPacActionPerformed

private void codigoPacKeyTyped(java.awt.event.KeyEvent evt) {GEN-FIRST:event_codigoPacKeyTyped
    // TODO add your handling code here:
    char cTeclaPresionada=evt.getKeyChar();
    // da click en ingresar al detectar la tecla ENTER.
    if(cTeclaPresionada==KeyEvent.VK_ENTER){
        //Ejecuta el botón (dar click)
        ingresarBtn.doClick();
    } // fin del if.
} //GEN-LAST:event_codigoPacKeyTyped

private void nombrePacKeyTyped(java.awt.event.KeyEvent evt) {GEN-FIRST:event_nombrePacKeyTyped
    // TODO add your handling code here:
    char cTeclaPresionada=evt.getKeyChar();
    // da click en ingresar al detectar la tecla ENTER.
    if(cTeclaPresionada==KeyEvent.VK_ENTER){
        //Ejecuta el botón (dar click)
        ingresarBtn.doClick();
    } // fin del if.
} //GEN-LAST:event_nombrePacKeyTyped

private void borrarCKeyTyped(java.awt.event.KeyEvent evt) {GEN-FIRST:event_borrarCKeyTyped
    // TODO add your handling code here:
    char cTeclaPresionada=evt.getKeyChar();
    // da click al detectar la tecla ENTER.
    if(cTeclaPresionada==KeyEvent.VK_ENTER){
        //Ejecuta el botón (dar click)
        borrarC.doClick();
    } // fin del if.
} //GEN-LAST:event_borrarCKeyTyped

```

```

private void borrarCActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_borrarCActionPerformed
    // TODO add your handling code here:
    codigoPac.setText(""); // Se borra el contenido del codigo.
    nombrePac.setText(""); // Se borra el contenido del nombre.
} //GEN-LAST:event_borrarCActionPerformed

private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_jMenuItem4ActionPerformed
    // TODO add your handling code here:
    ManualW guiaRapida=new ManualW();
    guiaRapida.setVisible(true);
    setTitle("Manual de Ayuda");
} //GEN-LAST:event_jMenuItem4ActionPerformed

private void mINIActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_mINIActionPerformed
    // TODO add your handling code here:
    MainW menu=new MainW();
    menu.setVisible(true); // Vuelve el menu principal
    dispose(); // Desaparece el informes.
} //GEN-LAST:event_mINIActionPerformed

private void mInfActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_mInfActionPerformed
    // TODO add your handling code here:
    // TODO add your handling code here:
    InformesW iw=new InformesW();
    iw.setVisible(true); // Hago visible la ventana de Informes.
    dispose(); // Desaparece el menu.
} //GEN-LAST:event_mInfActionPerformed

private void SalirActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_SalirActionPerformed
    // TODO add your handling code here:
    System.exit(0); // SE SALE DEL PROGRAMA.
} //GEN-LAST:event_SalirActionPerformed

private void cerrarSesionActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_cerrarSesionActionPerformed
    // TODO add your handling code here:
    Login log=new Login(); // Abre el log principal del sistema.
    log.setVisible(true); // Hace visible el log.
    dispose(); // SE OCULTA
} //GEN-LAST:event_cerrarSesionActionPerformed

private void mIngDatActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_mIngDatActionPerformed
    // TODO add your handling code here:
    IngresoPacientesW ipw=new IngresoPacientesW();
    ipw.setVisible(true); // Hago visible la ventana de pacientes.
    dispose(); // Para que se oculte el menu.
} //GEN-LAST:event_mIngDatActionPerformed

/**
 * @param args the command line arguments

```

```

*/
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
    * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(DatosPacienteW.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(DatosPacienteW.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(DatosPacienteW.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(DatosPacienteW.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    }
    //</editor-fold>

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

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JMenuItem Salir;
private javax.swing.JButton anterior;
private javax.swing.JButton borrarC;
private javax.swing.JMenuItem cerrarSesion;
private javax.swing.JTextField codigoPac;
private javax.swing.JButton ingresarBtn;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JMenu jMenu4;
private javax.swing.JMenu jMenu5;
private javax.swing.JMenuBar jMenuBar3;
private javax.swing.JMenuItem jMenuItem4;
private javax.swing.JMenuItem mINI;

```

```

private javax.swing.JMenuItem mInf;
private javax.swing.JMenuItem mIngDat;
public javax.swing.JTextField nombrePac;
private javax.swing.JMenu opciones;
// End of variables declaration//GEN-END:variables
}

```

#### INGRESOPACIENTESW.JAVA

```

package controlpac;
import java.awt.event.KeyEvent;
import javax.swing.ImageIcon;

```

```

public class IngresoPacientesW extends javax.swing.JFrame {

```

```

    /**
     * Creates new form IngresoPacientesW
     */
    public IngresoPacientesW() {
        initComponents();
        setIconImage(new ImageIcon(getClass().getResource("/imagenes/iconoCM.png")).getImage()); //Agrego el
        icono a la ventana
    }

```

```

        setSize(450,400); //tamaño de ventana
        setLocationRelativeTo(null); //que se muestre en el centro
        setResizable(false);
        setTitle("Ingreso de Datos"); // Le da un titulo a la ventana.
    }

```

```

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

```

```

        jLabel1 = new javax.swing.JLabel();
        datosPacienteBtn = new javax.swing.JButton();
        situacionPacBtn = new javax.swing.JButton();
        datosMedBtn = new javax.swing.JButton();
        ant = new javax.swing.JButton();
        jLabel2 = new javax.swing.JLabel();
        jMenuBar1 = new javax.swing.JMenuBar();
        jMenu2 = new javax.swing.JMenu();
        jMenuItem2 = new javax.swing.JMenuItem();
        jMenu3 = new javax.swing.JMenu();
        mINI = new javax.swing.JMenuItem();
        mInf = new javax.swing.JMenuItem();
        opciones = new javax.swing.JMenu();
        Salir = new javax.swing.JMenuItem();
        cerrarSesion = new javax.swing.JMenuItem();
    }

```

```

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
getContentPane().setLayout(null);

jLabel1.setFont(new java.awt.Font("Cambria", 0, 18)); // NOI18N
jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel1.setText("INGRESO DE DATOS");
jLabel1.setBorder(javax.swing.BorderFactory.createEtchedBorder());
getContentPane().add(jLabel1);
jLabel1.setBounds(10, 24, 424, 26);

datosPacienteBtn.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
datosPacienteBtn.setText("Datos del paciente");
datosPacienteBtn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        datosPacienteBtnActionPerformed(evt);
    }
});
datosPacienteBtn.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        datosPacienteBtnKeyTyped(evt);
    }
});
getContentPane().add(datosPacienteBtn);
datosPacienteBtn.setBounds(100, 90, 220, 47);

situacionPacBtn.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
situacionPacBtn.setText("Situación del paciente");
situacionPacBtn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        situacionPacBtnActionPerformed(evt);
    }
});
situacionPacBtn.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        situacionPacBtnKeyTyped(evt);
    }
});
getContentPane().add(situacionPacBtn);
situacionPacBtn.setBounds(100, 150, 220, 47);

datosMedBtn.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
datosMedBtn.setText("Datos del médico");
datosMedBtn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        datosMedBtnActionPerformed(evt);
    }
});
datosMedBtn.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        datosMedBtnKeyTyped(evt);
    }
});

```

```

getContentPane().add(datosMedBtn);
datosMedBtn.setBounds(100, 210, 220, 44);

ant.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
ant.setText("Anterior");
ant.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        antActionPerformed(evt);
    }
});
ant.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        antKeyTyped(evt);
    }
});
getContentPane().add(ant);
ant.setBounds(38, 289, 94, 37);

jLabel2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/imagenes/fondo.jpg"))); // NOI18N
getContentPane().add(jLabel2);
jLabel2.setBounds(0, 0, 520, 360);

jMenu2.setText("Ayuda");
jMenu2.setToolTipText("");

jMenuItem2.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_A,
java.awt.event.InputEvent.CTRL_MASK));
jMenuItem2.setText("Manual");
jMenuItem2.setToolTipText("Vista de guía rápida de uso.");
jMenuItem2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem2ActionPerformed(evt);
    }
});
jMenu2.add(jMenuItem2);

jMenuBar1.add(jMenu2);

jMenu3.setText("Ir a");

mINI.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_H,
java.awt.event.InputEvent.CTRL_MASK));
mINI.setText("Menu Inicio");
mINI.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        mINIActionPerformed(evt);
    }
});
jMenu3.add(mINI);

mInf.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_I,
java.awt.event.InputEvent.CTRL_MASK));
mInf.setText("Informes");

```



```

mInf.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        mInfActionPerformed(evt);
    }
});
jMenu3.add(mInf);

jMenuBar1.add(jMenu3);

opciones.setText("Opciones");

Salir.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_S,
java.awt.event.InputEvent.CTRL_MASK));
Salir.setText("Salir del Programa");
Salir.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        SalirActionPerformed(evt);
    }
});
opciones.add(Salir);

cerrarSesion.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_C,
java.awt.event.InputEvent.CTRL_MASK));
cerrarSesion.setText("Cerrar Sesión");
cerrarSesion.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cerrarSesionActionPerformed(evt);
    }
});
opciones.add(cerrarSesion);

jMenuBar1.add(opciones);

setJMenuBar(jMenuBar1);

pack();
} // </editor-fold> // GEN-END: initComponents

private void datosPacienteBtnActionPerformed(java.awt.event.ActionEvent evt) // GEN-
FIRST:event_datosPacienteBtnActionPerformed
// TODO add your handling code here:
DatosPacienteW dp=new DatosPacienteW();
dp.setVisible(true);
dispose(); // se oculta la ventana de ingreso de datos.
} // GEN-LAST:event_datosPacienteBtnActionPerformed

private void situacionPacBtnActionPerformed(java.awt.event.ActionEvent evt) // GEN-
FIRST:event_situacionPacBtnActionPerformed
// TODO add your handling code here:
SituacionPacienteW sp=new SituacionPacienteW();
sp.setVisible(true);
dispose(); // se oculta la ventana de ingreso de datos.
} // GEN-LAST:event_situacionPacBtnActionPerformed

```

```

private void datosMedBtnActionPerformed(java.awt.event.ActionEvent evt) {GEN-
FIRST:event_datosMedBtnActionPerformed
    // TODO add your handling code here:
    DatosMedicoW dm=new DatosMedicoW();
    dm.setVisible(true);
    dispose(); // se oculta la ventana de ingreso de datos.
} //GEN-LAST:event_datosMedBtnActionPerformed

private void antActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_antActionPerformed
    // TODO add your handling code here:
    MainW menu=new MainW();
    menu.setVisible(true); // Vuelve el menu principal
    dispose(); // Desaparece el ingreso de datos.

} //GEN-LAST:event_antActionPerformed

private void antKeyPressed(java.awt.event.KeyEvent evt) {GEN-FIRST:event_antKeyPressed
    // TODO add your handling code here:
    // Creamos un evento para que cuando se presione la tecla Enter sea como presionar el botón.

//*****
    char cTeclaPresionada=evt.getKeyChar();
    // da click al detectar la tecla ENTER.
    if(cTeclaPresionada==KeyEvent.VK_ENTER){
        //Ejecuta el botón (dar click)
        ant.doClick();
    } // fin del if.
} //GEN-LAST:event_antKeyPressed

private void datosPacienteBtnKeyPressed(java.awt.event.KeyEvent evt) {GEN-
FIRST:event_datosPacienteBtnKeyPressed
    // TODO add your handling code here:
    // Creamos un evento para que cuando se presione la tecla Enter sea como presionar el botón.

//*****
    char cTeclaPresionada=evt.getKeyChar();
    // da click al detectar la tecla ENTER.
    if(cTeclaPresionada==KeyEvent.VK_ENTER){
        //Ejecuta el botón (dar click)
        datosPacienteBtn.doClick();
    } // fin del if.
} //GEN-LAST:event_datosPacienteBtnKeyPressed

private void situacionPacBtnKeyPressed(java.awt.event.KeyEvent evt) {GEN-
FIRST:event_situacionPacBtnKeyPressed
    // TODO add your handling code here:
    // Creamos un evento para que cuando se presione la tecla Enter sea como presionar el botón.

//*****
    char cTeclaPresionada=evt.getKeyChar();
    // da click al detectar la tecla ENTER.
    if(cTeclaPresionada==KeyEvent.VK_ENTER){

```

```

        //Ejecuta el botón (dar click)
        situacionPacBtn.doClick();
    }// fin del if.
} //GEN-LAST:event_situacionPacBtnKeyTyped

private void datosMedBtnKeyTyped(java.awt.event.KeyEvent evt) { //GEN-FIRST:event_datosMedBtnKeyTyped
    // TODO add your handling code here:
    // Creamos un evento para que cuando se presione la tecla Enter sea como presionar el botón.

    /*******
    char cTeclaPresionada=evt.getKeyChar();
    // da click al detectar la tecla ENTER.
    if(cTeclaPresionada==KeyEvent.VK_ENTER){
        //Ejecuta el botón (dar click)
        datosMedBtn.doClick();
    }// fin del if.
} //GEN-LAST:event_datosMedBtnKeyTyped

private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_jMenuItem2ActionPerformed
    // TODO add your handling code here:
    ManualW guiaRapida=new ManualW();
    guiaRapida.setVisible(true);
    setTitle("Manual de Ayuda");
} //GEN-LAST:event_jMenuItem2ActionPerformed

private void mINIActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_mINIActionPerformed
    // TODO add your handling code here:
    MainW menu=new MainW();
    menu.setVisible(true); // Vuelve el menu principal
    dispose(); // Desaparece el informes.
} //GEN-LAST:event_mINIActionPerformed

private void SalirActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_SalirActionPerformed
    // TODO add your handling code here:
    System.exit(0); // SE SALE DEL PROGRAMA.
} //GEN-LAST:event_SalirActionPerformed

private void cerrarSesionActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_cerrarSesionActionPerformed
    // TODO add your handling code here:
    Login log=new Login(); // Abre el log principal del sistema.
    log.setVisible(true); // Hace visible el log.
    dispose();//SE OCULTA
} //GEN-LAST:event_cerrarSesionActionPerformed

private void mInfActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_mInfActionPerformed
    // TODO add your handling code here:
    // TODO add your handling code here:
    InformesW iw=new InformesW();
    iw.setVisible(true); // Hago visible la ventana de Informes.
    dispose(); // Desaparece el menu.
} //GEN-LAST:event_mInfActionPerformed

```

```

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(IngresoPacientesW.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(IngresoPacientesW.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(IngresoPacientesW.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(IngresoPacientesW.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    }
}
//</editor-fold>

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

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JMenuItem Salir;
private javax.swing.JButton ant;
private javax.swing.JMenuItem cerrarSesion;
private javax.swing.JButton datosMedBtn;
private javax.swing.JButton datosPacienteBtn;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JMenu jMenu2;
private javax.swing.JMenu jMenu3;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JMenuItem jMenuItem2;
private javax.swing.JMenuItem mINI;

```

```

private javax.swing.JMenuItem mInf;
private javax.swing.JMenu opciones;
private javax.swing.JButton situacionPacBtn;
// End of variables declaration//GEN-END:variables
}

```

#### LOGIN.JAVA

```

package controlpac;
import java.awt.Image;
import java.awt.Toolkit;
import java.awt.event.KeyEvent;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;

```

```

public class Login extends javax.swing.JFrame {

```

```

    String usuarioValido="admin"; // Este es el usuario valido del sistema.
    String contraseñaValida="admin"; // Esta es la contraseña valida del sistema.
    String usuario=""; // Para el ingreso de usuario.
    String contraseña=""; // Para el ingreso de contraseña.

```

```

/**

```

```

 * Creates new form Login

```

```

 */

```

```

public Login() {

```

```

    initComponents();

```

```

    setIconImage(new ImageIcon(getClass().getResource("/imagenes/iconoCM.png")).getImage()); //Agrego el
    icono a la ventana

```

```

    setSize(750,490);

```

```

    setLocationRelativeTo(null); // Para que el login aparezca en el centro de la pantalla.

```

```

    setResizable(false); // que no pueda agrandar la pantalla.

```

```

    setTitle("Identificación"); // Le da un titulo a la ventana.

```

```

}

```

```

public String getUsuario(){

```

```

    this.usuario=""; //La cadena Limpia.

```

```

    this.usuario=user.getText(); //Obtenemos el nombre de usuario ingresado.

```

```

    return usuario; // Devolvemos el nombre de usuario.

```

```

}

```

```

public String getPass(){

```

```

    this.contraseña=""; //Cadena de String vacía para luego guardar la contraseña ingresada.

```

```

    char [] pass=password.getPassword(); // Obtenemos la contraseña que ingresó el usuario.

```

```

    for(int i=0; i<pass.length;i++){ //Debemos realizar un for para copiar la pass ingresada a la cadena.

```

```

        this.contraseña+=pass[i]; //Se copia.

```

```

    }

```

```

    return contraseña;

```

```

}

```

```

/**

```

```

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

    jPanel1 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    user = new javax.swing.JTextField();
    password = new javax.swing.JPasswordField();
    ingresar = new javax.swing.JButton();
    jLabel5 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setIconImages(getIconImages());
    getContentPane().setLayout(null);

    jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Acceso al Sistema",
    javax.swing.border.TitledBorder.LEFT, javax.swing.border.TitledBorder.BOTTOM, new java.awt.Font("Candara", 0,
    12), java.awt.Color.darkGray)); // NOI18N

    jLabel1.setText("Usuario:");

    jLabel4.setText("Contraseña:");

    user.setToolTipText("Ingrese su nombre de usuario.");
    user.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyTyped(java.awt.event.KeyEvent evt) {
            userKeyTyped(evt);
        }
    });

    password.setToolTipText("Ingrese su contraseña.");
    password.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyTyped(java.awt.event.KeyEvent evt) {
            passwordKeyTyped(evt);
        }
    });

    ingresar.setText("Ingresar");
    ingresar.setToolTipText("Click para Ingresar al Sistema");
    ingresar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            ingresarActionPerformed(evt);
        }
    });
    ingresar.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyTyped(java.awt.event.KeyEvent evt) {
            ingresarKeyTyped(evt);
        }
    });
}

```

```

});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addComponent(ingresar)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(jLabel4)
                .addComponent(jLabel1))
            .addGap(16, 16, 16)
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                .addComponent(user, javax.swing.GroupLayout.DEFAULT_SIZE, 114, Short.MAX_VALUE)
                .addComponent(password))
            .addGap(20, 20, 20))
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel1)
            .addComponent(user, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel4)
            .addComponent(password, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.RELATED)
            .addComponent(ingresar)
            .addGap(16, 16, 16))
    );

jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(ingresar)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel1)
            .addComponent(user, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel4)
            .addComponent(password, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(ingresar)
        .addContainerGap()
    );

getContentPane().add(jPanel1);
jPanel1.setBounds(20, 150, 220, 120);

jLabel5.setIcon(new javax.swing.ImageIcon(getClass().getResource("/imagenes/portadaL.jpg"))); // NOI18N
getContentPane().add(jLabel5);
jLabel5.setBounds(0, -190, 820, 700);

pack();
}

private void ingresarActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_ingresarActionPerformed
    // TODO add your handling code here:
    getUsuario(); //Obtenemos el usuario.
    getPass(); //Obtenemos la contraseña.
    // Se muestra un mensaje si alguno de los campos está vacío.
}

```

```
if(this.usuario==" " || this.contraseña==""){
    JOptionPane.showMessageDialog(null, "Por favor, ingrese nombre de usuario y contraseña."); //Mensaje de Error.
}else{
    if(!this.usuario.equals(this.usuarioValido)){
        // Si el usuario y la contraseña no son válidos.
        if(!this.contraseña.equals(this.contraseñaValida))
            JOptionPane.showMessageDialog(null,"Usuario y Contraseña no son válidos. Vuelva a ingresar.");
        else // Sólo el usuario no es válido.
            JOptionPane.showMessageDialog(null, "El usuario no es correcto. Vuelva a ingresar.");
    }else{ // Sólo la contraseña no es válida.
        if(!this.contraseña.equals(this.contraseñaValida))
            JOptionPane.showMessageDialog(null,"Contraseña Incorrecta. Vuelva a ingresar.");
        else{ //Contraseña y usuario son válidas.
            MainW menu=new MainW();
            menu.setVisible(true); // hace visibla la ventana de menu principal.
            setTitle("Menú Principal"); // Le da un titulo a la ventana.
            menu.setLocationRelativeTo(null);
            dispose();
        }
    }
}
}
}
}
//GEN-LAST:event_ingresarActionPerformed
```



```

char cTeclaPresionada=evt.getKeyChar();
// da click al detectar la tecla ENTER.
if(cTeclaPresionada==KeyEvent.VK_ENTER){
    //Ejecuta el botón (dar click)
    ingresar.doClick();
} // fin del if.
} //GEN-LAST:event_ingresarKeyTyped

/**
 * @param args the command line arguments
 */

public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

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

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton ingresar;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JPanel jPanel1;
private javax.swing.JPasswordField password;

```

```

private javax.swing.JTextField user;
// End of variables declaration//GEN-END:variables
}

```

#### MAINW.JAVA

```

package controlpac;
import java.awt.Color;
import java.awt.event.KeyEvent;
import javax.swing.ImageIcon;

public class MainW extends javax.swing.JFrame {

    /**
     * Creates new form EnfermedadesMedW
     */
    public MainW() {

        initComponents();
        setIconImage(new ImageIcon(getClass().getResource("/imagenes/iconoCM.png")).getImage()); //Agrego el
        icono a la ventana
        setSize(575,518);
        setResizable(false);
        setLocationRelativeTo(null); // Para que aparezca en el centro de la pantalla.
        setTitle("Menú Principal"); // Le da un titulo a la ventana.

    }

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

        jLabel4 = new javax.swing.JLabel();
        informes = new javax.swing.JButton();
        salir = new javax.swing.JButton();
        ingreso = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();
        jMenuBar1 = new javax.swing.JMenuBar();
        jMenu2 = new javax.swing.JMenu();
        jMenuItem2 = new javax.swing.JMenuItem();
        jMenu3 = new javax.swing.JMenu();
        mIngDat = new javax.swing.JMenuItem();
        mInf = new javax.swing.JMenuItem();
        opciones = new javax.swing.JMenu();
        Salir = new javax.swing.JMenuItem();
        cerrarSesion = new javax.swing.JMenuItem();

    }

```

```

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setBackground(new java.awt.Color(255, 255, 255));
getContentPane().setLayout(null);

jLabel4.setFont(new java.awt.Font("Cambria", 0, 18)); // NOI18N
jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel4.setText("CONTROL DE PACIENTES");
jLabel4.setBorder(javax.swing.BorderFactory.createEtchedBorder());
getContentPane().add(jLabel4);
jLabel4.setBounds(10, 120, 550, 26);

informes.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
informes.setText("Informes");
informes.setToolTipText("Click para ver enfermedades y pacientes que atienden los médicos.");
informes.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        informesActionPerformed(evt);
    }
});
informes.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        informesKeyTyped(evt);
    }
});
getContentPane().add(informes);
informes.setBounds(40, 270, 180, 64);

salir.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
salir.setText("Salir");
salir.setToolTipText("Click para salir del programa.");
salir.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        salirActionPerformed(evt);
    }
});
salir.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        salirKeyTyped(evt);
    }
});
getContentPane().add(salir);
salir.setBounds(40, 360, 180, 64);

ingreso.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
ingreso.setText("Ingreso de Datos");
ingreso.setToolTipText("Click para agregar información sobre médicos y pacientes.");
ingreso.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ingresoActionPerformed(evt);
    }
});
ingreso.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {

```

```

        ingresoKeyTyped(evt);
    }
});
getContentPane().add(ingreso);
ingreso.setBounds(40, 180, 180, 64);

jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/imagenes/portada.jpg"))); // NOI18N
jLabel1.setText("jLabel1");
getContentPane().add(jLabel1);
jLabel1.setBounds(0, -20, 780, 640);

jMenu2.setText("Ayuda");
jMenu2.setToolTipText("Obtenga ayuda.");

jMenuItem2.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_A,
java.awt.event.InputEvent.CTRL_MASK));
jMenuItem2.setText("Manual");
jMenuItem2.setToolTipText("Vista de guía rápida de uso.");
jMenuItem2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem2ActionPerformed(evt);
    }
});
jMenu2.add(jMenuItem2);

jMenuBar1.add(jMenu2);

jMenu3.setText("Ir a");

mIngDat.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_D,
java.awt.event.InputEvent.CTRL_MASK));
mIngDat.setText("Ingreso de Datos");
mIngDat.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        mIngDatActionPerformed(evt);
    }
});
jMenu3.add(mIngDat);

mInf.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_I,
java.awt.event.InputEvent.CTRL_MASK));
mInf.setText("Informes");
mInf.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        mInfActionPerformed(evt);
    }
});
jMenu3.add(mInf);

jMenuBar1.add(jMenu3);

opciones.setText("Opciones");

```

```

        Salir.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_S,
java.awt.event.InputEvent.CTRL_MASK));
        Salir.setText("Salir del Programa");
        Salir.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                SalirActionPerformed(evt);
            }
        });
        opciones.add(Salir);

        cerrarSesion.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_C,
java.awt.event.InputEvent.CTRL_MASK));
        cerrarSesion.setText("Cerrar Sesión");
        cerrarSesion.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                cerrarSesionActionPerformed(evt);
            }
        });
        opciones.add(cerrarSesion);

        jMenuBar1.add(opciones);

        setJMenuBar(jMenuBar1);

        pack();
    } // </editor-fold> //GEN-END: initComponents

    private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_jMenuItem2ActionPerformed
        // TODO add your handling code here:
        ManualW guiaRapida=new ManualW();
        guiaRapida.setVisible(true);
        setTitle("Manual de Ayuda");
    } //GEN-LAST:event_jMenuItem2ActionPerformed

    private void mIngDatActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_mIngDatActionPerformed
        // TODO add your handling code here:
        IngresoPacientesW menu=new IngresoPacientesW();
        menu.setVisible(true); // Vuelve el menu principal
        dispose(); // Desaparece el informes.
    } //GEN-LAST:event_mIngDatActionPerformed

    private void SalirActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_SalirActionPerformed
        // TODO add your handling code here:
        System.exit(0); // SE SALE DEL PROGRAMA.
    } //GEN-LAST:event_SalirActionPerformed

    private void cerrarSesionActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_cerrarSesionActionPerformed
        // TODO add your handling code here:
        Login log=new Login(); // Abre el log principal del sistema.
        log.setVisible(true); // Hace visible el log.
    }

```

```

        dispose();//SE OCULTA
    }//GEN-LAST:event_cerrarSesionActionPerformed

    private void mInfActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_mInfActionPerformed
        // TODO add your handling code here:
        InformesW ipw=new InformesW();
        ipw.setVisible(true); // Hago visible la ventana de pacientes.
        dispose(); // Para que se oculte el menu.
    }//GEN-LAST:event_mInfActionPerformed

    private void ingresoActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_ingresoActionPerformed
        // TODO add your handling code here:
        IngresoPacientesW menu=new IngresoPacientesW();
        menu.setVisible(true); // Vuelve el menu principal
        dispose(); // Desaparece el informes.
    }//GEN-LAST:event_ingresoActionPerformed

    private void salirActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_salirActionPerformed
        // TODO add your handling code here:
        System.exit(0);//SALE DEL PROGRAMA
    }//GEN-LAST:event_salirActionPerformed

    private void informesActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_informesActionPerformed
        // TODO add your handling code here:
        InformesW ipw=new InformesW();
        ipw.setVisible(true); // Hago visible la ventana de pacientes.
        dispose(); // Para que se oculte el menu.
    }//GEN-LAST:event_informesActionPerformed

    private void ingresoKeyTyped(java.awt.event.KeyEvent evt) {//GEN-FIRST:event_ingresoKeyTyped
        // TODO add your handling code here:
        char cTeclaPresionada=evt.getKeyChar();
        // da click al detectar la tecla ENTER.
        if(cTeclaPresionada==KeyEvent.VK_ENTER){
            //Ejecuta el botón (dar click)
            ingreso.doClick();
        }// fin del if.
    }//GEN-LAST:event_ingresoKeyTyped

    private void informesKeyTyped(java.awt.event.KeyEvent evt) {//GEN-FIRST:event_informesKeyTyped
        // TODO add your handling code here:
        char cTeclaPresionada=evt.getKeyChar();
        // da click al detectar la tecla ENTER.
        if(cTeclaPresionada==KeyEvent.VK_ENTER){
            //Ejecuta el botón (dar click)
            informes.doClick();
        }// fin del if.
    }//GEN-LAST:event_informesKeyTyped

    private void salirKeyTyped(java.awt.event.KeyEvent evt) {//GEN-FIRST:event_salirKeyTyped
        // TODO add your handling code here:

```

```

        char cTeclaPresionada=evt.getKeyChar();
// da click al detectar la tecla ENTER.
if(cTeclaPresionada==KeyEvent.VK_ENTER){
    //Ejecuta el botón (dar click)
    salir.doClick();
} // fin del if.
} //GEN-LAST:event_salirKeyTyped

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(MainW.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(MainW.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(MainW.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(MainW.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
} //</editor-fold>

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

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JMenuItem salir;
private javax.swing.JMenuItem cerrarSesion;
private javax.swing.JButton informes;
private javax.swing.JButton ingreso;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel4;
private javax.swing.JMenu jMenu2;
private javax.swing.JMenu jMenu3;
private javax.swing.JMenuBar jMenuBar1;

```

```

private javax.swing.JMenuItem jMenuItem2;
private javax.swing.JMenuItem mInfi;
private javax.swing.JMenuItem mIngDat;
private javax.swing.JMenu opciones;
private javax.swing.JButton salir;
// End of variables declaration//GEN-END:variables
}

```

## MANUALW.JAVA

```

package controlpac;
import java.awt.event.KeyEvent;
import javax.swing.ImageIcon;

```

```

public class ManualW extends javax.swing.JFrame {

```

```

    /**
     * Creates new form ManualW
     */
    public ManualW() {
        initComponents();
        setIconImage(new ImageIcon(getClass().getResource("/imagenes/iconoCM.png")).getImage()); //Agrego el
        icono a la ventana
        setSize(400,450);
        //setLocationRelativeTo(null);
        setTitle("Manual Rápido");
        setDefaultCloseOperation(DISPOSE_ON_CLOSE); // Desctivo el botón cerrar (x) para evitar que se cierre el
        proyecto.
    }

```

```

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

```

```

        jLabel1 = new javax.swing.JLabel();
        jScrollPane1 = new javax.swing.JScrollPane();
        jTextPane1 = new javax.swing.JTextPane();
        cerrar = new javax.swing.JButton();
        jLabel2 = new javax.swing.JLabel();

```

```

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
        getContentPane().setLayout(null);

```

```

        jLabel1.setFont(new java.awt.Font("Cambria", 0, 18)); // NOI18N
        jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

```



```
jLabel1.setText("Guía Rápida para el Usuario");
jLabel1.setBorder(javax.swing.BorderFactory.createEtchedBorder());
getContentPane().add(jLabel1);
jLabel1.setBounds(10, 11, 380, 26);

jScrollPane1.setAutoscrolls(true);
jScrollPane1.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
jScrollPane1.setFocusable(false);
jScrollPane1.addMouseWheelListener(new java.awt.event.MouseWheelListener() {
    public void mouseWheelMoved(java.awt.event.MouseWheelEvent evt) {
        jScrollPane1MouseWheelMoved(evt);
    }
});

jTextPane1.setEditable(false);
jTextPane1.setBorder(javax.swing.BorderFactory.createEmptyBorder(1, 1, 1, 1));
jTextPane1.setFont(new java.awt.Font("Calibri", 0, 11)); // NOI18N
jTextPane1.setText("*****\n\t\t\t\t\tATAJOS\n\t\t\t\t\t\n*****\nn> ABRIR ESTE MANUAL DE AYUDA:\tCtrl + A\n> IR AL MENÚ INICIO:\tCtrl + H\n> IR A INFORMES:\tCtrl + I\n> IR AL INGRESO DE DATOS:\tCtrl + D\n> CERRAR SESION DE USUARIO:\tCtrl+C\n> SALIR DEL PROGRAMA:\tCtrl+S\n*****\nOPERACIONES\t\n*****\nn* Para agregar información sobre los pacientes o los médicos: \n\t\t\t\t\tseleccione el botón \"INGRESO DE DATOS\" del menú principal.\nn* Para ver las enfermedades o los pacientes que atienden\n los médicos: \n\t\t\t\t\tseleccione el botón \"INFORMES\" del menú principal.\n");
jScrollPane1.setViewportView(jTextPane1);

getContentPane().add(jScrollPane1);
jScrollPane1.setBounds(10, 43, 380, 290);

cerrar.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
cerrar.setText("Cerrar Manual");
cerrar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cerrarActionPerformed(evt);
    }
});
cerrar.addFocusListener(new java.awt.event.FocusAdapter() {
    public void focusGained(java.awt.event.FocusEvent evt) {
        cerrarFocusGained(evt);
    }
});
cerrar.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        cerrarKeyTyped(evt);
    }
});
getContentPane().add(cerrar);
cerrar.setBounds(10, 344, 170, 39);

jLabel2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/imagenes/fondo.jpg"))); // NOI18N
```

```

        getContentPane().add(jLabel2);
        jLabel2.setBounds(0, 0, 530, 460);

        pack();
    } // </editor-fold> // GEN-END: initComponents

    private void cerrarActionPerformed(java.awt.event.ActionEvent evt) { // GEN-FIRST: event_cerrarActionPerformed
        // TODO add your handling code here:
        dispose(); // Se oculta la ayuda.
    } // GEN-LAST: event_cerrarActionPerformed

    private void cerrarFocusGained(java.awt.event.FocusEvent evt) { // GEN-FIRST: event_cerrarFocusGained
        // TODO add your handling code here:
    } // GEN-LAST: event_cerrarFocusGained

    private void jScrollPane1MouseWheelMoved(java.awt.event.MouseWheelEvent evt) { // GEN-
FIRST: event_jScrollPane1MouseWheelMoved
        // TODO add your handling code here:
    } // GEN-LAST: event_jScrollPane1MouseWheelMoved

    private void cerrarKeyTyped(java.awt.event.KeyEvent evt) { // GEN-FIRST: event_cerrarKeyTyped
        // TODO add your handling code here:
        // Creamos un evento para que cuando se presione la tecla Enter sea como presionar el botón.

    } // GEN-LAST: event_cerrarKeyTyped

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        // <editor-fold defaultstate="collapsed" desc="Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(ManualW.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(ManualW.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
    }
}

```

```

    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(ManualW.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(ManualW.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
}
//</editor-fold>

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

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton cerrar;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTextPane jTextPane1;
// End of variables declaration//GEN-END:variables
}

```

#### PACIENTESXMedW.JAVA

```

package controlpac;
import java.awt.event.KeyEvent;
import java.io.DataInputStream;
import java.io.EOFException;
import java.io.FileInputStream;
import java.io.IOException;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;

public class PacientesXMedW extends javax.swing.JFrame {

    String codMed="";
    String nomMed="";

    String codm=""; // En estas variables vamos a guardar los datos de la búsqueda del medico.
    String nomm="";
    String espm="";
    String codp=""; // Estas variables que siguen nos ayudan en la búsqueda de datos del médico.
    String codme="";
    String enfp="";
    String codpa=""; // Nos sirven para la lectura de pacientes.
    String nomp="";
    String checkPaciente=""; //Para que no se un mismo paciente en la lista, ya que puede tener muchos
    diagnosticos.
    String lista=""; // Limpio la lista de pacientes a mostrar.
    /**

```

```

* Creates new form PacientesXMedW
*/
public PacientesXMedW() {
    initComponents();
    setIconImage(new ImageIcon(getClass().getResource("/imagenes/iconoCM.png")).getImage()); //Agrego el
    icono a la ventana
    setSize(575,518); //tamaño de ventana
    setResizable(false); // que no la puedan cerrar
    setLocationRelativeTo(null);
    setTitle("Informes"); // Le da un titulo a la ventana.
}

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

    jLabel1 = new javax.swing.JLabel();
    jPanel1 = new javax.swing.JPanel();
    jScrollPane1 = new javax.swing.JScrollPane();
    listaPac = new javax.swing.JTextArea();
    ant = new javax.swing.JButton();
    jLabel3 = new javax.swing.JLabel();
    nombreMed = new javax.swing.JTextField();
    jPanel2 = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    codigoMed = new javax.swing.JTextField();
    consultarBtn = new javax.swing.JButton();
    borrarC = new javax.swing.JButton();
    jLabel4 = new javax.swing.JLabel();
    jMenuBar1 = new javax.swing.JMenuBar();
    jMenu2 = new javax.swing.JMenu();
    jMenuItem2 = new javax.swing.JMenuItem();
    jMenu3 = new javax.swing.JMenu();
    mINI = new javax.swing.JMenuItem();
    mIngDat = new javax.swing.JMenuItem();
    mInf = new javax.swing.JMenuItem();
    opciones = new javax.swing.JMenu();
    Salir = new javax.swing.JMenuItem();
    cerrarSesion = new javax.swing.JMenuItem();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    jLabel1.setFont(new java.awt.Font("Cambria", 0, 18)); // NOI18N
    jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    jLabel1.setText("<html><u>Pacientes por médico</u></html>");
    getContentPane().add(jLabel1);
    jLabel1.setBounds(10, 11, 544, 26);

```

```

jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder("Pacientes atendidos:"));

listaPac.setEditable(false);
listaPac.setColumns(20);
listaPac.setRows(5);
jScrollPane1.setViewportViewView(listaPac);

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 481,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(10, 10, 10)
        )
);
jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 145, Short.MAX_VALUE)
        )
);

getContentPane().add(jPanel1);
jPanel1.setBounds(20, 215, 517, 169);

ant.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
ant.setText("Anterior");
ant.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        antActionPerformed(evt);
    }
});
ant.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        antKeyTyped(evt);
    }
});
getContentPane().add(ant);
ant.setBounds(20, 402, 120, 33);

jLabel3.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jLabel3.setText("Nombre del médico:");
getContentPane().add(jLabel3);
jLabel3.setBounds(253, 189, 123, 17);

nombreMed.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
nombreMed.setDisabledTextColor(new java.awt.Color(0, 0, 0));
nombreMed.setEnabled(false);
nombreMed.setSelectedTextColor(new java.awt.Color(0, 0, 0));
nombreMed.setSelectionColor(new java.awt.Color(204, 204, 204));
getContentPane().add(nombreMed);
nombreMed.setBounds(380, 188, 150, 21);

```

```

jPanel2.setBorder(javax.swing.BorderFactory.createEtchedBorder());

jLabel2.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jLabel2.setText("Código del médico:");

codigoMed.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
codigoMed.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        codigoMedKeyTyped(evt);
    }
});

consultarBtn.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
consultarBtn.setText("Consultar");
consultarBtn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        consultarBtnActionPerformed(evt);
    }
});
consultarBtn.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        consultarBtnKeyTyped(evt);
    }
});

borrarC.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
borrarC.setText("Borrar");
borrarC.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        borrarCActionPerformed(evt);
    }
});
borrarC.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        borrarCKeyTyped(evt);
    }
});

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(28, 28, 28)
            .addComponent(jLabel2)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(codigoMed, javax.swing.GroupLayout.PREFERRED_SIZE, 140,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(245, 245, Short.MAX_VALUE)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(0, 0, Short.MAX_VALUE)
                .addComponent(consultarBtn)

```



```

    }
    });
    jMenu3.add(mIni);

    mIngDat.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_D,
java.awt.event.InputEvent.CTRL_MASK));
    mIngDat.setText("Ingreso de Datos");
    mIngDat.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            mIngDatActionPerformed(evt);
        }
    });
    jMenu3.add(mIngDat);

    mInf.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_I,
java.awt.event.InputEvent.CTRL_MASK));
    mInf.setText("Informes");
    mInf.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            mInfActionPerformed(evt);
        }
    });
    jMenu3.add(mInf);

    jMenuBar1.add(jMenu3);

    opciones.setText("Opciones");

    Salir.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_S,
java.awt.event.InputEvent.CTRL_MASK));
    Salir.setText("Salir del Programa");
    Salir.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            SalirActionPerformed(evt);
        }
    });
    opciones.add(Salir);

    cerrarSesion.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_C,
java.awt.event.InputEvent.CTRL_MASK));
    cerrarSesion.setText("Cerrar Sesión");
    cerrarSesion.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            cerrarSesionActionPerformed(evt);
        }
    });
    opciones.add(cerrarSesion);

    jMenuBar1.add(opciones);

    setJMenuBar(jMenuBar1);

    pack();

```



```

} // </editor-fold> // GEN-END: initComponents
public void consulta3(){
    try{
        DataInputStream datopac = null;
        datopac = new DataInputStream(new FileInputStream(System.getProperty("user.home") + "\\datopac.txt"));
//abre el archivo de pacientes para lectura
        int sw1 = 1;
        while (sw1 != 0) {
            try { // lee el registro
                codpa = EncryptHelper.Desencriptar(datopac.readUTF());
                nompa = EncryptHelper.Desencriptar(datopac.readUTF());
                if (codpa.equals(codp) && checkPaciente.indexOf(codpa)==-1){ //compara el codigo del paciente de la
tabla "situpac" con el codigo del paciente de la tabla "datopac"
                    // Hacemos checkPaciente.indexOf(codpa)==-1 para evitar duplicados de pacientes mostrados en al lista.
Devuelve -1 si no existe en la lista.
                    checkPaciente= checkPaciente + "#" + codp; // uso el signo # para separar los codigos ya encontrados
para listar.
                    lista = lista + nompa + "\n"; //Acumulo en la lista
                }
            } catch (Exception e) {
                sw1 = 0;
            }
        }
    } catch (IOException ioe){
        JOptionPane.showMessageDialog(null, "No se ha encontrado información sobre el médico solicitado."); //
ERROR
    }
}

public void consulta2(){
    try{
        DataInputStream situpac = null; // Preparo para abrir el archivo de pacientes.
        situpac = new DataInputStream(new FileInputStream(System.getProperty("user.home") + "\\situpac.txt"));
//abre el archivo de situacion del paciente para lectura
        int sw = 1; // Bandera para nueva búsqueda.
        while (sw != 0) {
            try { // lee el registro
                codp = EncryptHelper.Desencriptar(situpac.readUTF());
                codme = EncryptHelper.Desencriptar(situpac.readUTF());
                enfpa = EncryptHelper.Desencriptar(situpac.readUTF());
                if (codme.equals(codMed)) { //compara el codigo medico de la tabla "datomed" con el de la tabla "situpac"
                    consulta3();
                }
            } catch (Exception e) {
                sw = 0;
            }
        }
    } catch (IOException ioe){
        JOptionPane.showMessageDialog(null, "No se ha encontrado información sobre el médico solicitado."); //
ERROR
    }
}

public void consultarPacientes(){

```

```

try {
    DataInputStream datomed = null; //Para luego abrir el archivo de medicos.
    datomed = new DataInputStream(new FileInputStream(System.getProperty("user.home") +
"\datomed.txt")); // abre el archivo de datos de los medicos para lectura
    boolean found = false; // Nos va a servir para saber si encontramos al médico.
    int sw = 1; //Bandera para la búsqueda.
    while (sw != 0) {
        try { // lee un registro
            codm = EncryptHelper.Desencriptar(datomed.readUTF());
            nomm = EncryptHelper.Desencriptar(datomed.readUTF());
            espm = EncryptHelper.Desencriptar(datomed.readUTF());
        } catch (Exception e) {
            sw = 0;
        }
        if (codm.equals(codMed)){ //compara el codigo extraido de la tabla "datomed" con el codigo ingresado
            found = true; // ENCONTRADO.
            nombreMed.setText(nomm); // Hacemos que aparezca el nombre del médico.
            consulta2(); // hago otra consulta sobre le archivo de situacion de pacientes.
        }
    }
    if (!found) { // si el medico no se encontro se informa
        JOptionPane.showMessageDialog(null, "No se ha encontrado información sobre el médico solicitado.");
    }else
        listaPac.setText(lista); //Mestro la lista
    } catch (IOException ioe) {
        JOptionPane.showMessageDialog(null, "No se ha encontrado información sobre el médico solicitado.");
    };
}

private void antActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_antActionPerformed
    // TODO add your handling code here:
    InformesW menu=new InformesW();
    menu.setVisible(true); // Vuelve el menu de informes
    dispose(); // Desaparece el informe de pacientes por medico.
} //GEN-LAST:event_antActionPerformed

private void antKeyTyped(java.awt.event.KeyEvent evt) {GEN-FIRST:event_antKeyTyped
    // TODO add your handling code here:
    // Creamos un evento para que cuando se presione la tecla Enter sea como presionar el botón.

    /*******
    char cTeclaPresionada=evt.getKeyChar();
    // da click al detectar la tecla ENTER.
    if(cTeclaPresionada==KeyEvent.VK_ENTER){
        //Ejecuta el botón (dar click)
        ant.doClick();
    } // fin del if.
    } //GEN-LAST:event_antKeyTyped

    private void borrarCActionPerformed(java.awt.event.ActionEvent evt) {GEN-
FIRST:event_borrarCActionPerformed
    // TODO add your handling code here:
    codigoMed.setText(""); // Limpio los datos.

```

```

nombreMed.setText("");
listaPac.setText("");

} //GEN-LAST:event_borrarCActionPerformed

private void borrarCKeyTyped(java.awt.event.KeyEvent evt) { //GEN-FIRST:event_borrarCKeyTyped
    // TODO add your handling code here:
    char cTeclaPresionada=evt.getKeyChar();
    // da click al detectar la tecla ENTER.
    if(cTeclaPresionada==KeyEvent.VK_ENTER){
        //Ejecuta el botón (dar click)
        borrarC.doClick();
    } // fin del if.
} //GEN-LAST:event_borrarCKeyTyped

private void consultarBtnKeyTyped(java.awt.event.KeyEvent evt) { //GEN-FIRST:event_consultarBtnKeyTyped
    // TODO add your handling code here:
    char cTeclaPresionada=evt.getKeyChar();
    // da click al detectar la tecla ENTER.
    if(cTeclaPresionada==KeyEvent.VK_ENTER){
        //Ejecuta el botón (dar click)
        consultarBtn.doClick();
    } // fin del if.
} //GEN-LAST:event_consultarBtnKeyTyped

private void consultarBtnActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_consultarBtnActionPerformed
    // TODO add your handling code here:
    this.codMed=codigoMed.getText(); // Sería un getCodMed();
    lista="";
    checkPaciente="";
    listaPac.setText(""); // limpio para mostrar uno nuevo.
    DatosMedicoW medico=new DatosMedicoW(codMed); // Creo un medico para usar el metodo que verifique
que existe.
    if(!medico.medicoExiste()){ //Si no existe.
        JOptionPane.showMessageDialog(null, "Código de médico inexistente."); //Muestro error.
    } else
        consultarPacientes(); // Existe el médico, enotnces muestro los pacientes si es que tiene.
} //GEN-LAST:event_consultarBtnActionPerformed

private void codigoMedKeyTyped(java.awt.event.KeyEvent evt) { //GEN-FIRST:event_codigoMedKeyTyped
    // TODO add your handling code here:
    // TODO add your handling code here:
    char cTeclaPresionada=evt.getKeyChar();
    // da click al detectar la tecla ENTER.
    if(cTeclaPresionada==KeyEvent.VK_ENTER){
        //Ejecuta el botón (dar click)
        consultarBtn.doClick();
    } // fin del if.
} //GEN-LAST:event_codigoMedKeyTyped

private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_jMenuItem2ActionPerformed

```

```

// TODO add your handling code here:
ManualW guiaRapida=new ManualW();
guiaRapida.setVisible(true);
setTitle("Manual de Ayuda");
} //GEN-LAST:event_jMenuItem2ActionPerformed

private void mINIActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_mINIActionPerformed
// TODO add your handling code here:
MainW menu=new MainW();
menu.setVisible(true); // Vuelve el menu principal
dispose(); // Desaparece el informes.
} //GEN-LAST:event_mINIActionPerformed

private void mInfActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_mInfActionPerformed
// TODO add your handling code here:
// TODO add your handling code here:
InformesW iw=new InformesW();
iw.setVisible(true); // Hago visible la ventana de Informes.
dispose(); // Desaparece el menu.
} //GEN-LAST:event_mInfActionPerformed

private void SalirActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_SalirActionPerformed
// TODO add your handling code here:
System.exit(0); // SE SALE DEL PROGRAMA.
} //GEN-LAST:event_SalirActionPerformed

private void cerrarSesionActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_cerrarSesionActionPerformed
// TODO add your handling code here:
Login log=new Login(); // Abre el log principal del sistema.
log.setVisible(true); // Hace visible el log.
dispose(); // SE OCULTA
} //GEN-LAST:event_cerrarSesionActionPerformed

private void mIngDatActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_mIngDatActionPerformed
// TODO add your handling code here:
IngresoPacientesW ipw=new IngresoPacientesW();
ipw.setVisible(true); // Hago visible la ventana de pacientes.
dispose(); // Para que se oculte el menu.
} //GEN-LAST:event_mIngDatActionPerformed

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

```

```

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

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

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JMenuItem Salir;
private javax.swing.JButton ant;
private javax.swing.JButton borrarC;
private javax.swing.JMenuItem cerrarSesion;
private javax.swing.JTextField codigoMed;
private javax.swing.JButton consultarBtn;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JMenu jMenu2;
private javax.swing.JMenu jMenu3;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JMenuItem jMenuItem2;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTextArea listaPac;
private javax.swing.JMenuItem mINI;
private javax.swing.JMenuItem mInf;
private javax.swing.JMenuItem mIngDat;
private javax.swing.JTextField nombreMed;
private javax.swing.JMenu opciones;
// End of variables declaration//GEN-END:variables

```

```
}
```

#### SITUACIONPACIENTEW.JAVA

```
package controlpac;
import java.awt.event.KeyEvent;
import java.io.DataOutputStream;
import java.io.FileOutputStream;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;

public class SituacionPacienteW extends javax.swing.JFrame {

    String codMed=""; // Para guardar el codigo de medico ingresado.
    String codPac=""; //Para guardar el codigo del paciente ingresado.
    String diagMed=""; // Para guardar el diagnostico del médico.
    /**
     * Creates new form SituacionPacienteW
     */
    public SituacionPacienteW() {
        initComponents();
        setIconImage(new ImageIcon(getClass().getResource("/imagenes/iconoCM.png")).getImage()); //Agrego el
        icono a la ventana
        setSize(575,518); //tamaño de ventana
        setResizable(false);
        setLocationRelativeTo(null); //La ventana centrada.
        setTitle("Ingreso de Datos"); // Le da un titulo a la ventana.
    }

    public String getCodMed(){
        this.codMed=codigoMed.getText(); // Se obtienen los datos ingresados.
        return codMed;
    }

    public String getCodPac(){
        this.codPac=codigoPac.getText(); // Se obtienen los datos ingresados.
        return codPac;
    }

    public String getDiagMed(){
        this.diagMed=diagnostico.getText(); // Se obtienen los datos ingresados.
        return diagMed;
    }

    public void guardarSitPac(){
        // Esta función guarda el diagnostico del paciente en el archivo de situaciones.
        try {
            DataOutputStream situpac = null;
            // abre el archivo de situacion del paciente para escritura con append habilitado
            situpac = new DataOutputStream(new FileOutputStream(System.getProperty("user.home") +
            "\\situpac.txt", true));
            // escribe el archivo con el registro
        }
    }
}
```

```

        situpac.writeUTF(EncryptHelper.Encriptar(codPac)); //Se guardan los datos.
        situpac.writeUTF(EncryptHelper.Encriptar(codMed));
        situpac.writeUTF(EncryptHelper.Encriptar(diagMed));
        JOptionPane.showMessageDialog(null, "Datos guardados de forma exitosa"); // Se ha guardado con éxito.
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Error al intentar abrir el archivo de situaciones de pacientes."); // Se
informa un error.
    }

}

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

    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    codigoPac = new javax.swing.JTextField();
    jLabel3 = new javax.swing.JLabel();
    codigoMed = new javax.swing.JTextField();
    jPanel1 = new javax.swing.JPanel();
    jScrollPane1 = new javax.swing.JScrollPane();
    diagnostico = new javax.swing.JTextArea();
    ingresarBtn = new javax.swing.JButton();
    ant = new javax.swing.JButton();
    borrarC = new javax.swing.JButton();
    jLabel4 = new javax.swing.JLabel();
    jMenuBar1 = new javax.swing.JMenuBar();
    jMenu2 = new javax.swing.JMenu();
    jMenuItem2 = new javax.swing.JMenuItem();
    Ira = new javax.swing.JMenu();
    mINI = new javax.swing.JMenuItem();
    mIngDat = new javax.swing.JMenuItem();
    mInf = new javax.swing.JMenuItem();
    opciones = new javax.swing.JMenu();
    Salir = new javax.swing.JMenuItem();
    cerrarSesion = new javax.swing.JMenuItem();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    jLabel1.setFont(new java.awt.Font("Cambria", 0, 18)); // NOI18N
    jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    jLabel1.setText("SITUACION DEL PACIENTE");
    jLabel1.setBorder(javax.swing.BorderFactory.createEtchedBorder());
    getContentPane().add(jLabel1);
    jLabel1.setBounds(10, 11, 541, 26);

    jLabel2.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N

```

```

jLabel2.setText("Código del paciente:");
getContentPane().add(jLabel2);
jLabel2.setBounds(40, 90, 124, 17);
getContentPane().add(codigoPac);
codigoPac.setBounds(170, 90, 99, 30);

jLabel3.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jLabel3.setText("Código del médico:");
getContentPane().add(jLabel3);
jLabel3.setBounds(310, 90, 117, 17);
getContentPane().add(codigoMed);
codigoMed.setBounds(433, 88, 99, 30);

jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder("Diagnóstico del médico"));

diagnostico.setColumns(20);
diagnostico.setLineWrap(true);
diagnostico.setRows(5);
diagnostico.setWrapStyleWord(true);
jScrollPane1.setViewportView(diagnostico);

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 509, Short.MAX_VALUE)
            .addGap(10, 10, 10)
        )
);
jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 157, Short.MAX_VALUE)
            .addGap(10, 10, 10)
        )
);

getContentPane().add(jPanel1);
jPanel1.setBounds(10, 177, 541, 202);

ingresarBtn.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
ingresarBtn.setText("Ingresar");
ingresarBtn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ingresarBtnActionPerformed(evt);
    }
});
ingresarBtn.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        ingresarBtnKeyTyped(evt);
    }
});

```



```

getContentPane().add(ingresarBtn);
ingresarBtn.setBounds(446, 398, 90, 31);

ant.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
ant.setText("Anterior");
ant.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        antActionPerformed(evt);
    }
});
ant.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        antKeyTyped(evt);
    }
});
getContentPane().add(ant);
ant.setBounds(24, 398, 100, 31);

borrarC.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
borrarC.setText("Borrar Campos");
borrarC.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        borrarCActionPerformed(evt);
    }
});
borrarC.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        borrarCKeyTyped(evt);
    }
});
getContentPane().add(borrarC);
borrarC.setBounds(311, 398, 130, 31);

jLabel4.setIcon(new javax.swing.ImageIcon(getClass().getResource("/imagenes/fondo.jpg"))); // NOI18N
getContentPane().add(jLabel4);
jLabel4.setBounds(0, 0, 680, 590);

jMenu2.setText("Ayuda");
jMenu2.setTooltipText("");

jMenuItem2.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_A,
java.awt.event.InputEvent.CTRL_MASK));
jMenuItem2.setText("Manual");
jMenuItem2.setTooltipText("Vista de guía rápida de uso.");
jMenuItem2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem2ActionPerformed(evt);
    }
});
jMenu2.add(jMenuItem2);

jMenuBar1.add(jMenu2);

```

```

Ira.setText("Ira");

mINI.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_H,
java.awt.event.InputEvent.CTRL_MASK));
mINI.setText("Menu Inicio");
mINI.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        mINIActionPerformed(evt);
    }
});
Ira.add(mINI);

mIngDat.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_D,
java.awt.event.InputEvent.CTRL_MASK));
mIngDat.setText("Ingreso de Datos");
mIngDat.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        mIngDatActionPerformed(evt);
    }
});
Ira.add(mIngDat);

mInf.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_I,
java.awt.event.InputEvent.CTRL_MASK));
mInf.setText("Informes");
mInf.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        mInfActionPerformed(evt);
    }
});
Ira.add(mInf);

jMenuBar1.add(Ira);

opciones.setText("Opciones");

Salir.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_S,
java.awt.event.InputEvent.CTRL_MASK));
Salir.setText("Salir del Programa");
Salir.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        SalirActionPerformed(evt);
    }
});
opciones.add(Salir);

cerrarSesion.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_C,
java.awt.event.InputEvent.CTRL_MASK));
cerrarSesion.setText("Cerrar Sesión");
cerrarSesion.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cerrarSesionActionPerformed(evt);
    }
}

```

```

});
opciones.add(cerrarSesion);

jMenuBar1.add(opciones);

setJMenuBar(jMenuBar1);

pack();
} // </editor-fold> // GEN-END: initComponents

private void antActionPerformed(java.awt.event.ActionEvent evt) // GEN-FIRST:event_antActionPerformed
// TODO add your handling code here:
IngresoPacientesW menu=new IngresoPacientesW();
menu.setVisible(true); // Vuelve el menu de ingreso de datos
dispose(); // Desaparece el ingreso de diagnostico.
} // GEN-LAST:event_antActionPerformed

private void antKeyPressed(java.awt.event.KeyEvent evt) // GEN-FIRST:event_antKeyPressed
// TODO add your handling code here:
// Creamos un evento para que cuando se presione la tecla Enter sea como presionar el botón.

//*****
char cTeclaPresionada=evt.getKeyChar();
// da click al detectar la tecla ENTER.
if(cTeclaPresionada==KeyEvent.VK_ENTER){
//Ejecuta el botón (dar click)
ant.doClick();
} // fin del if.
} // GEN-LAST:event_antKeyPressed

private void ingresarBtnKeyPressed(java.awt.event.KeyEvent evt) // GEN-FIRST:event_ingresarBtnKeyPressed
// TODO add your handling code here:
// Creamos un evento para que cuando se presione la tecla Enter sea como presionar el botón.

//*****
char cTeclaPresionada=evt.getKeyChar();
// da click al detectar la tecla ENTER.
if(cTeclaPresionada==KeyEvent.VK_ENTER){
//Ejecuta el botón (dar click)
ingresarBtn.doClick();
} // fin del if.
} // GEN-LAST:event_ingresarBtnKeyPressed

private void borrarCKeyPressed(java.awt.event.KeyEvent evt) // GEN-FIRST:event_borrarCKeyPressed
// TODO add your handling code here:
char cTeclaPresionada=evt.getKeyChar();
// da click al detectar la tecla ENTER.
if(cTeclaPresionada==KeyEvent.VK_ENTER){
//Ejecuta el botón (dar click)
borrarC.doClick();
} // fin del if.
} // GEN-LAST:event_borrarCKeyPressed

```

```

private void borrarCActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_borrarCActionPerformed
    // TODO add your handling code here:
    codigoPac.setText(""); //Limpio todos los campos.
    codigoMed.setText("");
    diagnostico.setText("");
} //GEN-LAST:event_borrarCActionPerformed

private void ingresarBtnActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_ingresarBtnActionPerformed
    // TODO add your handling code here:
    int errores=0; //Para contabilizar los errores.
    String msj=""; //Para mostrar mensajes de error.
    getCodPac(); //Obtengo los datos ingresados.
    getCodMed();
    getDiagMed();
    DatosPacienteW paciente=new DatosPacienteW(this.codPac); //Simulo haber creado un paciente para usar
los métodos de paciente.
    DatosMedicoW medico=new DatosMedicoW(this.codMed); // Igual que lo anterior, el objetivo es usar los
métodos de médico.
    if(!paciente.ExistePac()){ //Si el paciente no existe.
        msj="El código de paciente ingresado no existe. ";
        errores++; // Hay un error.
    }
    if(!medico.medicoExiste()){ // Si el médico no existe.
        msj= msj + "El código del médico ingresado no existe. ";
        errores++; //Hay un error.
    }
    if(this.diagMed.equals("")){
        msj= msj + "Por favor, ingrese el diagnóstico.";
        errores++; //Hay un error.
    }
    if(errores!=0)
        JOptionPane.showMessageDialog(null, msj); // Había errores, por lo tanto se muestra un mensaje y no se
guarda.
    else
        guardarSitPac(); //Sin errores entonces se guardan los datos.
} //GEN-LAST:event_ingresarBtnActionPerformed

private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_jMenuItem2ActionPerformed
    // TODO add your handling code here:
    ManualW guiaRapida=new ManualW();
    guiaRapida.setVisible(true);
    setTitle("Manual de Ayuda");
} //GEN-LAST:event_jMenuItem2ActionPerformed

private void mINIActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_mINIActionPerformed
    // TODO add your handling code here:
    MainW menu=new MainW();
    menu.setVisible(true); // Vuelve el menu principal
    dispose(); // Desaparece el informes.
} //GEN-LAST:event_mINIActionPerformed

```

```

private void mInfActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_mInfActionPerformed
    // TODO add your handling code here:
    // TODO add your handling code here:
    InformesW iw=new InformesW();
    iw.setVisible(true); // Hago visible la ventana de Informes.
    dispose(); // Desaparece el menu.
}GEN-LAST:event_mInfActionPerformed

private void SalirActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_SalirActionPerformed
    // TODO add your handling code here:
    System.exit(0); // SE SALE DEL PROGRAMA.
}GEN-LAST:event_SalirActionPerformed

private void cerrarSesionActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_cerrarSesionActionPerformed
    // TODO add your handling code here:
    Login log=new Login(); // Abre el log principal del sistema.
    log.setVisible(true); // Hace visible el log.
    dispose();//SE OCULTA
}GEN-LAST:event_cerrarSesionActionPerformed

private void mIngDatActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_mIngDatActionPerformed
    // TODO add your handling code here:
    IngresoPacientesW ipw=new IngresoPacientesW();
    ipw.setVisible(true); // Hago visible la ventana de pacientes.
    dispose(); // Para que se oculte el menu.
    dispose(); // Desaparece el informes.
}GEN-LAST:event_mIngDatActionPerformed

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(SituacionPacienteW.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(SituacionPacienteW.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
}

```

```

        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(SituacionPacienteW.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(SituacionPacienteW.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        }
    }
    //</editor-fold>

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

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JMenu Ira;
private javax.swing.JMenuItem Salir;
private javax.swing.JButton ant;
private javax.swing.JButton borrarC;
private javax.swing.JMenuItem cerrarSesion;
private javax.swing.JTextField codigoMed;
private javax.swing.JTextField codigoPac;
private javax.swing.JTextArea diagnostico;
private javax.swing.JButton ingresarBtn;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JMenu jMenu2;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JMenuItem jMenuItem2;
private javax.swing.JPanel jPanel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JMenuItem mINI;
private javax.swing.JMenuItem mInf;
private javax.swing.JMenuItem mIngDat;
private javax.swing.JMenu opciones;
// End of variables declaration//GEN-END:variables
}

```

#### [BDATO.A.JAVA](#)

```

package controlpac;
import java.io.*;
import javax.swing.JOptionPane;

public class bdatoa {

    public static void main(String [] args) /*throws Exception*/ {
        Login log=new Login(); // Abre el log principal del sistema.
    }
}

```

```

        log.setVisible(true); // Hace visible el log.
    }
}

```

#### DATOSMEDICOW.JAVA

```

package controlpac;
import java.awt.event.KeyEvent;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;

public class DatosMedicoW extends javax.swing.JFrame {

    String codigoMed=""; //Para guardar el nombre del medico ingresado.
    String nombreMed=""; //Para guardar el codigo del medico ingresado.
    String especialidadMed=""; //Para guardar la especialidad del medico ingresado.
    /**
     * Creates new form DatosMedicoW
     */
    public DatosMedicoW() { // Constructor 1
        initComponents();
        setIconImage(new ImageIcon(getClass().getResource("/imagenes/iconoCM.png")).getImage()); //Agrego el
        icono a la ventana
        setSize(500,400);
        setLocationRelativeTo(null); // En el centro la ventana.
        setResizable(false); // Anula poder maximizar la ventana.
        setTitle("Ingreso de Datos"); // Le da un titulo a la ventana.
    }

    public DatosMedicoW(String codigo) { // Constructor 2 para crear un medico desde otras clases
        this.codigoMed=codigo;
        //Sólo cargamos el dato básico para poder usar los metodos de medico.
    }

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

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        codigo = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        nombre = new javax.swing.JTextField();
    }
}

```

```

anterior = new javax.swing.JButton();
ingresarBtn = new javax.swing.JButton();
borrarC = new javax.swing.JButton();
jLabel4 = new javax.swing.JLabel();
especialidad = new javax.swing.JTextField();
jLabel5 = new javax.swing.JLabel();
jMenuBar1 = new javax.swing.JMenuBar();
jMenu2 = new javax.swing.JMenu();
jMenuItem2 = new javax.swing.JMenuItem();
jMenu3 = new javax.swing.JMenu();
mINI = new javax.swing.JMenuItem();
mIngDat = new javax.swing.JMenuItem();
mInf = new javax.swing.JMenuItem();
opciones = new javax.swing.JMenu();
Salir = new javax.swing.JMenuItem();
cerrarSesion = new javax.swing.JMenuItem();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setBackground(new java.awt.Color(255, 255, 255));
getContentPane().setLayout(null);

jLabel1.setFont(new java.awt.Font("Cambria", 0, 18)); // NOI18N
jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel1.setText("DATOS DEL MÉDICO");
jLabel1.setBorder(javax.swing.BorderFactory.createEtchedBorder());
getContentPane().add(jLabel1);
jLabel1.setBounds(10, 11, 457, 26);

jLabel2.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jLabel2.setText("Código:");
getContentPane().add(jLabel2);
jLabel2.setBounds(53, 84, 54, 17);

codigo.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
codigo.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        codigoKeyTyped(evt);
    }
});
getContentPane().add(codigo);
codigo.setBounds(125, 83, 93, 21);

jLabel3.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jLabel3.setText("Nombre:");
getContentPane().add(jLabel3);
jLabel3.setBounds(53, 123, 54, 17);

nombre.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
nombre.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        nombreKeyTyped(evt);
    }
});

```



```

getContentPane().add(nombre);
nombre.setBounds(125, 122, 157, 21);

anterior.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
anterior.setText("Anterior");
anterior.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        anteriorActionPerformed(evt);
    }
});
anterior.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        anteriorKeyTyped(evt);
    }
});
getContentPane().add(anterior);
anterior.setBounds(26, 253, 110, 33);

ingresarBtn.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
ingresarBtn.setText("Ingresar");
ingresarBtn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ingresarBtnActionPerformed(evt);
    }
});
ingresarBtn.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        ingresarBtnKeyTyped(evt);
    }
});
getContentPane().add(ingresarBtn);
ingresarBtn.setBounds(350, 200, 120, 30);

borrarC.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
borrarC.setText("Borrar Campos");
borrarC.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        borrarCActionPerformed(evt);
    }
});
borrarC.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        borrarCKeyTyped(evt);
    }
});
getContentPane().add(borrarC);
borrarC.setBounds(170, 200, 170, 30);

jLabel4.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jLabel4.setText("Especialidad:");
getContentPane().add(jLabel4);
jLabel4.setBounds(30, 162, 77, 17);

```

```

especialidad.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
especialidad.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        especialidadKeyTyped(evt);
    }
});
getContentPane().add(especialidad);
especialidad.setBounds(125, 161, 157, 21);

jLabel5.setIcon(new javax.swing.ImageIcon(getClass().getResource("/imagenes/fondo.jpg"))); // NOI18N
getContentPane().add(jLabel5);
jLabel5.setBounds(0, 0, 520, 350);

jMenu2.setText("Ayuda");
jMenu2.setToolTipText("");

jMenuItem2.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_A,
java.awt.event.InputEvent.CTRL_MASK));
jMenuItem2.setText("Manual");
jMenuItem2.setToolTipText("Vista de guía rápida de uso.");
jMenuItem2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem2ActionPerformed(evt);
    }
});
jMenu2.add(jMenuItem2);

jMenuBar1.add(jMenu2);

jMenu3.setText("Ir a");

mINI.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_H,
java.awt.event.InputEvent.CTRL_MASK));
mINI.setText("Menu Inicio");
mINI.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        mINIActionPerformed(evt);
    }
});
jMenu3.add(mINI);

mIngDat.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_D,
java.awt.event.InputEvent.CTRL_MASK));
mIngDat.setText("Ingreso de Datos");
mIngDat.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        mIngDatActionPerformed(evt);
    }
});
jMenu3.add(mIngDat);

mInf.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_I,
java.awt.event.InputEvent.CTRL_MASK));

```

```

mInf.setText("Informes");
mInf.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        mInfActionPerformed(evt);
    }
});
jMenu3.add(mInf);

jMenuBar1.add(jMenu3);

opciones.setText("Opciones");

Salir.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_S,
java.awt.event.InputEvent.CTRL_MASK));
Salir.setText("Salir del Programa");
Salir.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        SalirActionPerformed(evt);
    }
});
opciones.add(Salir);

cerrarSesion.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_C,
java.awt.event.InputEvent.CTRL_MASK));
cerrarSesion.setText("Cerrar Sesión");
cerrarSesion.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cerrarSesionActionPerformed(evt);
    }
});
opciones.add(cerrarSesion);

jMenuBar1.add(opciones);

setJMenuBar(jMenuBar1);

pack();
} // </editor-fold> // GEN-END: initComponents

public String getCodigoMed(){
    this.codigoMed=codigo.getText(); //Guarda el codigo ingresado.
    return codigoMed;
}

public String getNombreMed(){
    this.nombreMed=nombre.getText(); //Guarda el nombre Ingresado.
    return nombreMed;
}

public String getEspecialidadMed(){
    this.especialidadMed=especialidad.getText();
    return especialidadMed;
}

```

```

private static boolean soloLetras(String cadena){
    if(cadena.matches("[áéíóúña-zÑÁÉÍÓÚA-Z][áéíóúña-zÁÉÍÓÚÑA-z ]*")) // Se valida en esta función que no puede
    ser vacía la cadena.
        return true; //Son letras.
    else
        return false; //No son letras.
}

private static boolean alfanumerico(String cadena){
    if(cadena.matches("[0-9a-zA-z]+")) // Tampoco puede ser vacía una cadena.
        return true; // Es alfanumerico
    else
        return false; //No es alfanumerico
}

public int validarDatosMed(){
    String mensaje="";
    int ret=0; // comienza siendo valido.
    if(!alfanumerico(codigoMed)){ // El codigo del paciente debe ser alfanumerico sin espacios.
        mensaje="El código no es válido o no se ha ingresado. "; //Se guarda mensaje de aviso.
        ret++; //Ret se incrementa porque ya no es valido.
    }
    if(!soloLetras(nombreMed)){ // El nombre del paciente sólo debe contener letras.
        mensaje= mensaje + "El nombre del médico posee caracteres incorrectos o no se ha ingresado. "; //Se guarda
mensaje de aviso.
        ret++; // Ret deja de ser valido.
    }
    if(!soloLetras(especialidadMed)){ // El nombre del paciente sólo debe contener letras.
        mensaje= mensaje + "La especialidad del médico posee caracteres incorrectos o no se ha ingresado"; //Se
guarda mensaje de aviso.
        ret++; // Ret deja de ser valido.
    }
    if(ret!=0)
        JOptionPane.showMessageDialog(null, mensaje);
    return ret; // Correctos o Incorrectos.
}

public boolean medicoExiste() {
    int sw1 = 1;
    boolean found = false; //Contiene el resultado de la busqueda.
    try {
        DataInputStream datomedInput = null; // Luego abrimos el archivo de medicos.
        datomedInput = new DataInputStream(new FileInputStream(System.getProperty("user.home") +
"\datomed.txt")); // abre el archivo de medicos para lectura
        while (sw1 != 0) { //Bandera para la busqueda.
            String codm = EncryptHelper.Desencriptar(datomedInput.readUTF()); // lee el registro
            datomedInput.readUTF();
            datomedInput.readUTF();
            if (codm.equals(codigoMed)) {
                /*compara el codigo digitado con el codigo del medico de la
                tabla "datomed" */
                found = true; //ENCONTRADO
            }
        }
    }
}

```

```

        }
    }
    } catch (Exception e) {
        sw1 = 0;
    }
    return found; // Encontrado o no encontrado.
}

public void grabarDatosMed(){
    try { // Esta función se encarga de grabar los datos en el archivo de medicos.
        DataOutputStream datomed = null; // abre el archivo para escritura con append habilitado
        datomed = new DataOutputStream(new FileOutputStream(System.getProperty("user.home") +
"\\datomed.txt", true));
        //escribe el archivo con los datos ingresados
        datomed.writeUTF(EncryptHelper.Encriptar(codigoMed));
        datomed.writeUTF(EncryptHelper.Encriptar(nombreMed));
        datomed.writeUTF(EncryptHelper.Encriptar(especialidadMed));
        JOptionPane.showMessageDialog(null, "Datos guardados de forma exitosa."); // Se han guardado
correctamente los datos.
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Error al intentar abrir el archivo de médicos"); // Error inesperado
al abrir el arhivo.
    };
}
}

```

```

private void anteriorActionPerformed(java.awt.event.ActionEvent evt) {GEN-
FIRST:event_anteriorActionPerformed
    // TODO add your handling code here:
    IngresoPacientesW menu=new IngresoPacientesW();
    menu.setVisible(true); // Vuelve el menu de ingreso de datos
    dispose(); // Desaparece el ingreso de datos de medico.
} //GEN-LAST:event_anteriorActionPerformed

private void ingresarBtnKeyTyped(java.awt.event.KeyEvent evt) {GEN-FIRST:event_ingresarBtnKeyTyped
    // TODO add your handling code here:
    // Creamos un evento para que cuando se presione la tecla Enter sea como presionar el botón.

//*****
    char cTeclaPresionada=evt.getKeyChar();
    // da click al detectar la tecla ENTER.
    if(cTeclaPresionada==KeyEvent.VK_ENTER){
        //Ejecuta el botón (dar click)
        ingresarBtn.doClick();
    } // fin del if.
} //GEN-LAST:event_ingresarBtnKeyTyped

private void anteriorKeyTyped(java.awt.event.KeyEvent evt) {GEN-FIRST:event_anteriorKeyTyped
    // TODO add your handling code here:
    // Creamos un evento para que cuando se presione la tecla Enter sea como presionar el botón.

//*****

```

```

        char cTeclaPresionada=evt.getKeyChar();
        // da click al detectar la tecla ENTER.
        if(cTeclaPresionada==KeyEvent.VK_ENTER){
            //Ejecuta el botón (dar click)
            anterior.doClick();
        }// fin del if.
    }//GEN-LAST:event_anteriorKeyTyped

    private void borrarCActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_borrarCActionPerformed
        // TODO add your handling code here:
        codigo.setText(""); // Se borra el contenido del codigo del medico.
        nombre.setText(""); // Se borra el contenido del nombre de medico.
        especialidad.setText(""); // Se borra el contenido de la especialidad.
    }//GEN-LAST:event_borrarCActionPerformed

    private void borrarCKeyTyped(java.awt.event.KeyEvent evt) { //GEN-FIRST:event_borrarCKeyTyped
        // TODO add your handling code here:
        char cTeclaPresionada=evt.getKeyChar();
        // da click al detectar la tecla ENTER.
        if(cTeclaPresionada==KeyEvent.VK_ENTER){
            //Ejecuta el botón (dar click)
            borrarC.doClick();
        }// fin del if.
    }//GEN-LAST:event_borrarCKeyTyped

    private void ingresarBtnActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_ingresarBtnActionPerformed
        // TODO add your handling code here:
        getCodigoMed(); //Obtenemos el codigo del medico ingresado.
        getNombreMed(); // Obtenemos el nombre del medico ingresado.
        getEspecialidadMed(); // Obtenemos la especialidad del medico.
        if(validarDatosMed()==0){ // Si los datos estan bien ingresados, verifico la existencia del medico.
            if(!medicoExiste()) // Si el medico no está registrado en el archivo entonces lo agrego.
                grabarDatosMed(); //Se guardan los datos ingresados en el archivo.
            else
                JOptionPane.showMessageDialog(null, "El código del médico ya existe."); //Se informa que el medico ya existe.
        }
    }//GEN-LAST:event_ingresarBtnActionPerformed

    private void especialidadKeyTyped(java.awt.event.KeyEvent evt) { //GEN-FIRST:event_especialidadKeyTyped
        // TODO add your handling code here:
        char cTeclaPresionada=evt.getKeyChar();
        // da click al detectar la tecla ENTER.
        if(cTeclaPresionada==KeyEvent.VK_ENTER){
            //Ejecuta el botón ingresar datos (dar click)
            ingresarBtn.doClick();
        }// fin del if.
    }//GEN-LAST:event_especialidadKeyTyped

    private void nombreKeyTyped(java.awt.event.KeyEvent evt) { //GEN-FIRST:event_nombreKeyTyped
        // TODO add your handling code here:

```

```

        char cTeclaPresionada=evt.getKeyChar();
        // da click al detectar la tecla ENTER.
        if(cTeclaPresionada==KeyEvent.VK_ENTER){
            //Ejecuta el botón ingresar datos(dar click)
            ingresarBtn.doClick();
        }// fin del if.
    }//GEN-LAST:event_nombreKeyTyped

private void codigoKeyTyped(java.awt.event.KeyEvent evt) { //GEN-FIRST:event_codigoKeyTyped
    // TODO add your handling code here:
    char cTeclaPresionada=evt.getKeyChar();
    // da click al detectar la tecla ENTER.
    if(cTeclaPresionada==KeyEvent.VK_ENTER){
        //Ejecuta el botón ingresar datos(dar click)
        ingresarBtn.doClick();
    }// fin del if.
} //GEN-LAST:event_codigoKeyTyped

private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_jMenuItem2ActionPerformed
    // TODO add your handling code here:
    ManualW guiaRapida=new ManualW();
    guiaRapida.setVisible(true);
    setTitle("Manual de Ayuda");
} //GEN-LAST:event_jMenuItem2ActionPerformed

private void mINIActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_mINIActionPerformed
    // TODO add your handling code here:
    MainW menu=new MainW();
    menu.setVisible(true); // Vuelve el menu principal
    dispose(); // Desaparece el informes.
} //GEN-LAST:event_mINIActionPerformed

private void SalirActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_SalirActionPerformed
    // TODO add your handling code here:
    System.exit(0); // SE SALE DEL PROGRAMA.
} //GEN-LAST:event_SalirActionPerformed

private void cerrarSesionActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_cerrarSesionActionPerformed
    // TODO add your handling code here:
    Login log=new Login(); // Abre el log principal del sistema.
    log.setVisible(true); // Hace visible el log.
    dispose();//SE OCULTA
} //GEN-LAST:event_cerrarSesionActionPerformed

private void mInfActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_mInfActionPerformed
    // TODO add your handling code here:
    // TODO add your handling code here:
    InformesW iw=new InformesW();
    iw.setVisible(true); // Hago visible la ventana de Informes.
    dispose(); // Desaparece el menu.
} //GEN-LAST:event_mInfActionPerformed

```

```

private void mIngDatActionPerformed(java.awt.event.ActionEvent evt) {GEN-
FIRST:event_mIngDatActionPerformed
    // TODO add your handling code here:
        IngresoPacientesW ipw=new IngresoPacientesW();
        ipw.setVisible(true); // Hago visible la ventana de pacientes.
        dispose(); // Para que se oculte el menu.
    }GEN-LAST:event_mIngDatActionPerformed

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(DatosMedicoW.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(DatosMedicoW.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(DatosMedicoW.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(DatosMedicoW.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    }
    //</editor-fold>

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

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JMenuItem Salir;
private javax.swing.JButton anterior;
private javax.swing.JButton borrarC;
private javax.swing.JMenuItem cerrarSesion;

```



```

private javax.swing.JTextField codigo;
private javax.swing.JTextField especialidad;
private javax.swing.JButton ingresarBtn;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JMenu jMenu2;
private javax.swing.JMenu jMenu3;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JMenuItem jMenuItem2;
private javax.swing.JMenuItem mINI;
private javax.swing.JMenuItem mInf;
private javax.swing.JMenuItem mIngDat;
private javax.swing.JTextField nombre;
private javax.swing.JMenu opciones;
// End of variables declaration//GEN-END:variables
}

```

#### INFORMESW.JAVA

```
package controlpac;
```

```
import java.awt.event.KeyEvent;
import javax.swing.ImageIcon;
```

```
public class InformesW extends javax.swing.JFrame {
```

```

/**
 * Creates new form InformesW
 */
public InformesW() {
    initComponents();
    setIconImage(new ImageIcon(getClass().getResource("/imagenes/iconoCM.png")).getImage()); //Agrego el
icono a la ventana
    setSize(450,400);//tamaño de ventana
    setResizable(false); // Anula poder maximizar la ventana.
    setLocationRelativeTo(null);
    setTitle("Informes"); // Le da un titulo a la ventana.
}

```

```

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

    jLabel4 = new javax.swing.JLabel();
    jPanel1 = new javax.swing.JPanel();

```

```

listadoPacXMedBtn = new javax.swing.JButton();
EnfermedadesBtn = new javax.swing.JButton();
anterior = new javax.swing.JButton();
jLabel1 = new javax.swing.JLabel();
jMenuBar1 = new javax.swing.JMenuBar();
jMenu2 = new javax.swing.JMenu();
jMenuItem2 = new javax.swing.JMenuItem();
jMenu3 = new javax.swing.JMenu();
mINI = new javax.swing.JMenuItem();
mIngDat = new javax.swing.JMenuItem();
opciones = new javax.swing.JMenu();
Salir = new javax.swing.JMenuItem();
cerrarSesion = new javax.swing.JMenuItem();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
getContentPane().setLayout(null);

jLabel4.setFont(new java.awt.Font("Cambria", 0, 18)); // NOI18N
jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel4.setText("CONTROL DE PACIENTES");
jLabel4.setBorder(javax.swing.BorderFactory.createEtchedBorder());
getContentPane().add(jLabel4);
jLabel4.setBounds(30, 30, 390, 26);

jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "INFORMES",
javax.swing.border.TitledBorder.CENTER, javax.swing.border.TitledBorder.DEFAULT_POSITION));
jPanel1.setFont(new java.awt.Font("Cambria", 0, 12)); // NOI18N

listadoPacXMedBtn.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
listadoPacXMedBtn.setText("Listado de pacientes por médico");
listadoPacXMedBtn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        listadoPacXMedBtnActionPerformed(evt);
    }
});
listadoPacXMedBtn.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        listadoPacXMedBtnKeyTyped(evt);
    }
});

EnfermedadesBtn.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
EnfermedadesBtn.setText("Enfermedades que atiende cada médico");
EnfermedadesBtn.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        EnfermedadesBtnActionPerformed(evt);
    }
});
EnfermedadesBtn.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        EnfermedadesBtnKeyTyped(evt);
    }
});

```

```

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()
            .addContainerGap(54, Short.MAX_VALUE)
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                .addComponent(EnfermedadesBtn, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(listadoPacXMedBtn, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addGap(45, 45, 45))
        );
jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addContainerGap()
                .addComponent(listadoPacXMedBtn, javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 26, Short.MAX_VALUE)
                .addComponent(EnfermedadesBtn, javax.swing.GroupLayout.PREFERRED_SIZE, 36,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())
            );

getContentPane().add(jPanel1);
jPanel1.setBounds(30, 90, 390, 143);

anterior.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
anterior.setText("Anterior");
anterior.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        anteriorActionPerformed(evt);
    }
});
anterior.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyPressed(java.awt.event.KeyEvent evt) {
        anteriorKeyPressed(evt);
    }
});
getContentPane().add(anterior);
anterior.setBounds(30, 260, 86, 40);

jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/imagenes/fondo.jpg"))); // NOI18N
getContentPane().add(jLabel1);
jLabel1.setBounds(0, 0, 610, 460);

jMenu2.setText("Ayuda");
jMenu2.setTooltipText("");

 jMenuItem2.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_A,
java.awt.event.InputEvent.CTRL_MASK));

```

```

jMenuItem2.setText("Manual");
jMenuItem2.setToolTipText("Vista de guía rápida de uso.");
jMenuItem2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jMenuItem2ActionPerformed(evt);
    }
});
jMenu2.add(jMenuItem2);

jMenuBar1.add(jMenu2);

jMenu3.setText("Ir a");

mINI.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_H,
java.awt.event.InputEvent.CTRL_MASK));
mINI.setText("Menu Inicio");
mINI.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        mINIActionPerformed(evt);
    }
});
jMenu3.add(mINI);

mIngDat.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_D,
java.awt.event.InputEvent.CTRL_MASK));
mIngDat.setText("Ingreso de Datos");
mIngDat.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        mIngDatActionPerformed(evt);
    }
});
jMenu3.add(mIngDat);

jMenuBar1.add(jMenu3);

opciones.setText("Opciones");

Salir.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_S,
java.awt.event.InputEvent.CTRL_MASK));
Salir.setText("Salir del Programa");
Salir.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        SalirActionPerformed(evt);
    }
});
opciones.add(Salir);

cerrarSesion.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_C,
java.awt.event.InputEvent.CTRL_MASK));
cerrarSesion.setText("Cerrar Sesión");
cerrarSesion.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cerrarSesionActionPerformed(evt);
    }
});

```

```

    }
});
opciones.add(cerrarSesion);

jMenuBar1.add(opciones);

setJMenuBar(jMenuBar1);

pack();
} // </editor-fold> // GEN-END: initComponents

private void anteriorActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_anteriorActionPerformed
    MainW menu=new MainW();
    menu.setVisible(true); // Vuelve el menu principal
    dispose(); // Desaparece el informes.
} // GEN-LAST:event_anteriorActionPerformed

private void listadoPacXMedBtnActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_listadoPacXMedBtnActionPerformed
    // TODO add your handling code here:
    PacientesXMedW pxm=new PacientesXMedW();
    pxm.setVisible(true);
    dispose(); // Se oculta la ventana de informes.
} // GEN-LAST:event_listadoPacXMedBtnActionPerformed

private void EnfermedadesBtnActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_EnfermedadesBtnActionPerformed
    // TODO add your handling code here:
    EnfermedadesMedW em=new EnfermedadesMedW();
    em.setVisible(true);
    dispose(); // se oculta la ventana de informes.
} // GEN-LAST:event_EnfermedadesBtnActionPerformed

private void listadoPacXMedBtnKeyTyped(java.awt.event.KeyEvent evt) { // GEN-
FIRST:event_listadoPacXMedBtnKeyTyped
    // TODO add your handling code here:
    // Creamos un evento para que cuando se presione la tecla Enter sea como presionar el botón.

//*****
    char cTeclaPresionada=evt.getKeyChar();
    // da click al detectar la tecla ENTER.
    if(cTeclaPresionada==KeyEvent.VK_ENTER){
        //Ejecuta el botón (dar click)
        listadoPacXMedBtn.doClick();
    } // fin del if.
} // GEN-LAST:event_listadoPacXMedBtnKeyTyped

private void EnfermedadesBtnKeyTyped(java.awt.event.KeyEvent evt) { // GEN-
FIRST:event_EnfermedadesBtnKeyTyped
    // TODO add your handling code here:
    // Creamos un evento para que cuando se presione la tecla Enter sea como presionar el botón.

```

```

//*****
char cTeclaPresionada=evt.getKeyChar();
// da click al detectar la tecla ENTER.
if(cTeclaPresionada==KeyEvent.VK_ENTER){
    //Ejecuta el botón (dar click)
    EnfermedadesBtn.doClick();
} // fin del if.
} //GEN-LAST:event_EnfermedadesBtnKeyTyped

private void anteriorKeyTyped(java.awt.event.KeyEvent evt) { //GEN-FIRST:event_anteriorKeyTyped
    // TODO add your handling code here:
    // Creamos un evento para que cuando se presione la tecla Enter sea como presionar el botón.

//*****
char cTeclaPresionada=evt.getKeyChar();
// da click al detectar la tecla ENTER.
if(cTeclaPresionada==KeyEvent.VK_ENTER){
    //Ejecuta el botón (dar click)
    anterior.doClick();
} // fin del if.
} //GEN-LAST:event_anteriorKeyTyped

private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_jMenuItem2ActionPerformed
    // TODO add your handling code here:
    ManualW guiaRapida=new ManualW();
    guiaRapida.setVisible(true);
    setTitle("Manual de Ayuda");
} //GEN-LAST:event_jMenuItem2ActionPerformed

private void mINIActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_mINIActionPerformed
    // TODO add your handling code here:
    MainW menu=new MainW();
    menu.setVisible(true); // Vuelve el menu principal
    dispose(); // Desaparece el informes.
} //GEN-LAST:event_mINIActionPerformed

private void SalirActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_SalirActionPerformed
    // TODO add your handling code here:
    System.exit(0); // SE SALE DEL PROGRAMA.
} //GEN-LAST:event_SalirActionPerformed

private void cerrarSesionActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_cerrarSesionActionPerformed
    // TODO add your handling code here:
    Login log=new Login(); // Abre el log principal del sistema.
    log.setVisible(true); // Hace visible el log.
    dispose(); // SE OCULTA
} //GEN-LAST:event_cerrarSesionActionPerformed

private void mIngDatActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_mIngDatActionPerformed

```

```

// TODO add your handling code here:
IngresoPacientesW ipw=new IngresoPacientesW();
ipw.setVisible(true); // Hago visible la ventana de pacientes.
dispose(); // Para que se oculte el menu.
} //GEN-LAST:event_mIngDatActionPerformed

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(InformesW.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(InformesW.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(InformesW.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(InformesW.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
} //</editor-fold>

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

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton EnfermedadesBtn;
private javax.swing.JMenuItem Salir;
private javax.swing.JButton anterior;
private javax.swing.JMenuItem cerrarSesion;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel4;
private javax.swing.JMenu jMenu2;
private javax.swing.JMenu jMenu3;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JMenuItem jMenuItem2;
private javax.swing.JPanel jPanel1;

```

```

private javax.swing.JButton listadoPacXMedBtn;
private javax.swing.JMenuItem mINI;
private javax.swing.JMenuItem mIngDat;
private javax.swing.JMenu opciones;
// End of variables declaration//GEN-END:variables
}

```

#### ENFERMEDADESMEDW.JAVA

```

package controlpac;
import java.awt.event.KeyEvent;
import java.io.DataInputStream;
import java.io.EOFException;
import java.io.FileInputStream;
import java.io.IOException;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;

```

```

public class EnfermedadesMedW extends javax.swing.JFrame {

```

```

    /**
     * Creates new form MainWindow
     */
    String codMed="";
    String nomMed="";

```

```

    String codm=""; // En estas variables vamos a guardar los datos de la búsqueda del medico.
    String nomm="";
    String espm="";
    String codp=""; // Estas variables que siguen nos ayudan en la búsqueda de datos del médico.
    String codme="";
    String enfpa="";
    String codpa=""; // Nos sirven para la lectura de pacientes.
    String nompa="";
    String checkEnfermedades=""; //Para que no se muestre una misma enfermedad en la lista, ya que puede haber
    muchos diagnosticos de esa enfermedad.
    String lista=""; // Limpio la lista de enfermedades a mostrar.

```

```

    public EnfermedadesMedW() {
        initComponents();
        setIconImage(new ImageIcon(getClass().getResource("/imagenes/iconoCM.png")).getImage()); //Agrego el
        icono a la ventana
        setSize(350,480); //tamaño de ventana
        setResizable(false);
        setLocationRelativeTo(null); //centro
        setTitle("Informes"); // Le da un titulo a la ventana.
    }

```

```

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.

```



```

*/
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN: initComponents
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    nombreMed = new javax.swing.JTextField();
    jPanel1 = new javax.swing.JPanel();
    jScrollPane1 = new javax.swing.JScrollPane();
    listaEnf = new javax.swing.JTextArea();
    anterior = new javax.swing.JButton();
    jPanel2 = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    codigoMed = new javax.swing.JTextField();
    consultarBtn = new javax.swing.JButton();
    borrarC = new javax.swing.JButton();
    jLabel4 = new javax.swing.JLabel();
    jMenuBar1 = new javax.swing.JMenuBar();
    jMenu2 = new javax.swing.JMenu();
    jMenuItem2 = new javax.swing.JMenuItem();
    jMenu3 = new javax.swing.JMenu();
    mINI = new javax.swing.JMenuItem();
    mIngDat = new javax.swing.JMenuItem();
    mInf = new javax.swing.JMenuItem();
    opciones = new javax.swing.JMenu();
    Salir = new javax.swing.JMenuItem();
    cerrarSesion = new javax.swing.JMenuItem();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    getContentPane().setLayout(null);

    jLabel1.setFont(new java.awt.Font("Cambria", 0, 14)); // NOI18N
    jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    jLabel1.setText("<html><u>Enfermedades que atiende cada médico</u></html>");
    getContentPane().add(jLabel1);
    jLabel1.setBounds(10, 11, 290, 26);

    jLabel3.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
    jLabel3.setText("Nombre del médico:");
    getContentPane().add(jLabel3);
    jLabel3.setBounds(19, 141, 119, 15);

    nombreMed.setEditable(false);
    nombreMed.setDisabledTextColor(new java.awt.Color(0, 0, 0));
    nombreMed.setEnabled(false);
    nombreMed.setSelectedTextColor(new java.awt.Color(0, 0, 0));
    nombreMed.setSelectionColor(new java.awt.Color(153, 153, 153));
    getContentPane().add(nombreMed);
    nombreMed.setBounds(156, 139, 144, 20);

    jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder("Enfermedades que atiende:"));

```

```

listaEnf.setEditable(false);
listaEnf.setColumns(20);
listaEnf.setRows(5);
jScrollPane1.setViewportViewView(listaEnf);

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 294, Short.MAX_VALUE)
);
jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 175, Short.MAX_VALUE)
);

getContentPane().add(jPanel1);
jPanel1.setBounds(10, 170, 310, 199);

anterior.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
anterior.setText("Anterior");
anterior.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        anteriorActionPerformed(evt);
    }
});
anterior.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        anteriorKeyTyped(evt);
    }
});
getContentPane().add(anterior);
anterior.setBounds(10, 375, 100, 30);

jPanel2.setBorder(javax.swing.BorderFactory.createEtchedBorder());

jLabel2.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
jLabel2.setText("Código del médico:");

codigoMed.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        codigoMedActionPerformed(evt);
    }
});
codigoMed.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyTyped(java.awt.event.KeyEvent evt) {
        codigoMedKeyTyped(evt);
    }
});

consultarBtn.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
consultarBtn.setText("Consultar");
consultarBtn.addActionListener(new java.awt.event.ActionListener() {

```

```

        public void actionPerformed(java.awt.event.ActionEvent evt) {
            consultarBtnActionPerformed(evt);
        }
    });
    consultarBtn.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyTyped(java.awt.event.KeyEvent evt) {
            consultarBtnKeyTyped(evt);
        }
    });

    borrarC.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
    borrarC.setText("Borrar");
    borrarC.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            borrarCActionPerformed(evt);
        }
    });
    borrarC.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyTyped(java.awt.event.KeyEvent evt) {
            borrarCKeyTyped(evt);
        }
    });

    javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
    jPanel2.setLayout(jPanel2Layout);
    jPanel2Layout.setHorizontalGroup(
        jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(10, 10, 10)
                .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(codigoMed, javax.swing.GroupLayout.PREFERRED_SIZE, 124,
                        javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGroup(jPanel2Layout.createSequentialGroup()
                        .addGap(10, 10, 10)
                        .addComponent(consultarBtn)
                        .addGap(10, 10, 10)
                        .addComponent(borrarC, javax.swing.GroupLayout.PREFERRED_SIZE, 78,
                            javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addContainerGap(10, Short.MAX_VALUE))
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(10, 10, 10)
                .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jLabel2)
                    .addComponent(codigoMed, javax.swing.GroupLayout.PREFERRED_SIZE,
                        javax.swing.GroupLayout.PREFERRED_SIZE,
                        javax.swing.GroupLayout.PREFERRED_SIZE))
                .addContainerGap(10, Short.MAX_VALUE))
    );
    jPanel2Layout.setVerticalGroup(
        jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(10, 10, 10)
                .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(codigoMed, javax.swing.GroupLayout.PREFERRED_SIZE,
                        javax.swing.GroupLayout.PREFERRED_SIZE,
                        javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(consultarBtn)
                    .addComponent(borrarC, javax.swing.GroupLayout.PREFERRED_SIZE,
                        javax.swing.GroupLayout.PREFERRED_SIZE,
                        javax.swing.GroupLayout.PREFERRED_SIZE))
                .addContainerGap(10, Short.MAX_VALUE))
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(10, 10, 10)
                .addComponent(jLabel2)
                .addContainerGap(10, Short.MAX_VALUE))
    );

```

```

        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(consultarBtn)
            .addComponent(borrarC))
        .addContainerGap(16, Short.MAX_VALUE)
    );

    getContentPane().add(jPanel2);
    jPanel2.setBounds(19, 48, 300, 85);

    jLabel4.setIcon(new javax.swing.ImageIcon(getClass().getResource("/imagenes/fondo.jpg"))); // NOI18N
    getContentPane().add(jLabel4);
    jLabel4.setBounds(0, 0, 380, 440);

    jMenu2.setText("Ayuda");
    jMenu2.setToolTipText("");

    jMenuItem2.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_A,
        java.awt.event.InputEvent.CTRL_MASK));
    jMenuItem2.setText("Manual");
    jMenuItem2.setToolTipText("Vista de guía rápida de uso.");
    jMenuItem2.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jMenuItem2ActionPerformed(evt);
        }
    });
    jMenu2.add(jMenuItem2);

    jMenuBar1.add(jMenu2);

    jMenu3.setText("Ir a");

    mINI.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_H,
        java.awt.event.InputEvent.CTRL_MASK));
    mINI.setText("Menu Inicio");
    mINI.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            mINIActionPerformed(evt);
        }
    });
    jMenu3.add(mINI);

    mIngDat.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_D,
        java.awt.event.InputEvent.CTRL_MASK));
    mIngDat.setText("Ingreso de Datos");
    mIngDat.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            mIngDatActionPerformed(evt);
        }
    });
    jMenu3.add(mIngDat);

    mInf.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_I,
        java.awt.event.InputEvent.CTRL_MASK));

```

```

mInf.setText("Informes");
mInf.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        mInfActionPerformed(evt);
    }
});
jMenu3.add(mInf);

jMenuBar1.add(jMenu3);

opciones.setText("Opciones");

Salir.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_S,
java.awt.event.InputEvent.CTRL_MASK));
Salir.setText("Salir del Programa");
Salir.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        SalirActionPerformed(evt);
    }
});
opciones.add(Salir);

cerrarSesion.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_C,
java.awt.event.InputEvent.CTRL_MASK));
cerrarSesion.setText("Cerrar Sesión");
cerrarSesion.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        cerrarSesionActionPerformed(evt);
    }
});
opciones.add(cerrarSesion);

jMenuBar1.add(opciones);

setJMenuBar(jMenuBar1);

pack();
} // </editor-fold> // GEN-END: initComponents

public void consulta2(){
try{
    DataInputStream situpac = null; // Preparo para abrir el archivo de pacientes.
    situpac = new DataInputStream(new FileInputStream(System.getProperty("user.home") + "\\situpac.txt"));
    // abre el archivo de situacion del paciente para lectura
    int sw = 1; // Bandera para nueva búsqueda.
    while (sw != 0) {
        try { // lee el registro
            codp = EncryptHelper.Desencriptar(situpac.readUTF());
            codme = EncryptHelper.Desencriptar(situpac.readUTF());
            enfp = EncryptHelper.Desencriptar(situpac.readUTF());
            if (codme.equals(codMed) && checkEnfermedades.indexOf(enfp)==-1) { // compara el codigo medico de la
                tabla "datomed" con el de la tabla "situpac"
            }
        }
    }
}
}

```

```

        // Hacemos checkPaciente.indexOf(codpa)==-1 para evitar duplicados de pacientes mostrados en al lista.
        Devuelve -1 si no existe en la lista.
        checkEnfermedades= checkEnfermedades + "#" + enf; // uso el signo # para separar las enfermedades ya
        encontrados para listar.
        lista = lista + enf + "\n"; //Acumulo en la lista
    }
} catch (Exception e) {
    sw = 0;
}
}
} catch (IOException ioe){
    JOptionPane.showMessageDialog(null, "No se ha encontrado información sobre el médico solicitado."); //
ERROR
}
}
}

public void consultarEnfermedades(){

    try {
        DataInputStream datomed = null; //Para luego abrir el archivo de medicos.
        datomed = new DataInputStream(new FileInputStream(System.getProperty("user.home") +
        "\\datomed.txt")); // abre el archivo de datos de los medicos para lectura
        boolean found = false; // Nos va a servir para saber si encontramos al médico.
        int sw = 1; //Bandera para la búsqueda.
        while (sw != 0) {
            try { // lee un registro
                codm = EncryptHelper.Desencriptar(datomed.readUTF());
                nomm = EncryptHelper.Desencriptar(datomed.readUTF());
                espm = EncryptHelper.Desencriptar(datomed.readUTF());
            } catch (Exception e) {
                sw = 0;
            }
            if (codm.equals(codMed)){ //compara el codigo extraido de la tabla "datomed" con el codigo ingresado
                found = true; // ENCONTRADO.
                nombreMed.setText(nomm); // Hacemos que aparezca el nombre del médico.
                consulta2(); // hago otra consulta sobre le archivo de situacion de pacientes.
            }
        }
        if (!found) { // si el medico no se encontro se informa
            JOptionPane.showMessageDialog(null, "No se ha encontrado información sobre el médico solicitado.");
        } else
            listaEnf.setText(lista); //Mestro la lista
    } catch (IOException ioe) {
        JOptionPane.showMessageDialog(null, "No se ha encontrado información sobre el médico solicitado.");
    };
}
}

```

```

private void anteriorActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_anteriorActionPerformed
    // TODO add your handling code here:
    InformesW menu=new InformesW();
    menu.setVisible(true); // Vuelve el menu de informes
    dispose(); // Desaparece el informe de pacientes por medico.
}

```

```

} //GEN-LAST:event_anteriorActionPerformed

private void anteriorKeyTyped(java.awt.event.KeyEvent evt) { //GEN-FIRST:event_anteriorKeyTyped
    // TODO add your handling code here:
        // Creamos un evento para que cuando se presione la tecla Enter sea como presionar el botón.
    //*****
    char cTeclaPresionada=evt.getKeyChar();
    // da click al detectar la tecla ENTER.
    if(cTeclaPresionada==KeyEvent.VK_ENTER){
        //Ejecuta el botón (dar click)
        anterior.doClick();
    } // fin del if.
} //GEN-LAST:event_anteriorKeyTyped

private void consultarBtnActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_consultarBtnActionPerformed
    // TODO add your handling code here:
    this.codMed=codigoMed.getText(); // Sería un getCodMed();
    lista="";
    checkEnfermedades="";
    listaEnf.setText("");
    DatosMedicoW medico=new DatosMedicoW(codMed); // Creo un medico para usar el metodo que verifique
que existe.
    if(!medico.medicoExiste()){ //Si no existe.
        JOptionPane.showMessageDialog(null, "Código de médico inexistente."); //Muestro error.
    }else
        consultarEnfermedades(); // Existe el médico, entonces muestro las enfermedades que atiende.
} //GEN-LAST:event_consultarBtnActionPerformed

private void consultarBtnKeyTyped(java.awt.event.KeyEvent evt) { //GEN-FIRST:event_consultarBtnKeyTyped
    // TODO add your handling code here:
    char cTeclaPresionada=evt.getKeyChar();
    // da click al detectar la tecla ENTER.
    if(cTeclaPresionada==KeyEvent.VK_ENTER){
        //Ejecuta el botón (dar click)
        consultarBtn.doClick();
    } // fin del if.
} //GEN-LAST:event_consultarBtnKeyTyped

private void borrarCActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_borrarCActionPerformed
    // TODO add your handling code here:
    codigoMed.setText(""); // Limpio los datos.
    nombreMed.setText("");
    listaEnf.setText("");

} //GEN-LAST:event_borrarCActionPerformed

private void borrarCKeyTyped(java.awt.event.KeyEvent evt) { //GEN-FIRST:event_borrarCKeyTyped
    // TODO add your handling code here:
    char cTeclaPresionada=evt.getKeyChar();
    // da click al detectar la tecla ENTER.
    if(cTeclaPresionada==KeyEvent.VK_ENTER){

```

```

        //Ejecuta el botón (dar click)
        borrarC.doClick();
    } // fin del if.
} //GEN-LAST:event_borrarCKeyTyped

private void codigoMedActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_codigoMedActionPerformed
    // TODO add your handling code here:
} //GEN-LAST:event_codigoMedActionPerformed

private void codigoMedKeyTyped(java.awt.event.KeyEvent evt) { //GEN-FIRST:event_codigoMedKeyTyped
    // TODO add your handling code here:
    // TODO add your handling code here:
    char cTeclaPresionada=evt.getKeyChar();
    // da click al detectar la tecla ENTER.
    if(cTeclaPresionada==KeyEvent.VK_ENTER){
        //Ejecuta el botón (dar click)
        consultarBtn.doClick();
    } // fin del if.
} //GEN-LAST:event_codigoMedKeyTyped

private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_jMenuItem2ActionPerformed
    // TODO add your handling code here:
    ManualW guiaRapida=new ManualW();
    guiaRapida.setVisible(true);
    setTitle("Manual de Ayuda");
} //GEN-LAST:event_jMenuItem2ActionPerformed

private void mINIActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_mINIActionPerformed
    // TODO add your handling code here:
    MainW menu=new MainW();
    menu.setVisible(true); // Vuelve el menu principal
    dispose(); // Desaparece el informes.
} //GEN-LAST:event_mINIActionPerformed

private void mInfActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_mInfActionPerformed
    // TODO add your handling code here:
    // TODO add your handling code here:
    InformesW iw=new InformesW();
    iw.setVisible(true); // Hago visible la ventana de Informes.
    dispose(); // Desaparece el menu.
} //GEN-LAST:event_mInfActionPerformed

private void SalirActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_SalirActionPerformed
    // TODO add your handling code here:
    System.exit(0); // SE SALE DEL PROGRAMA.
} //GEN-LAST:event_SalirActionPerformed

private void cerrarSesionActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_cerrarSesionActionPerformed
    // TODO add your handling code here:
    Login log=new Login(); // Abre el log principal del sistema.

```



```

        log.setVisible(true); // Hace visible el log.
        dispose();//SE OCULTA
    }//GEN-LAST:event_cerrarSesionActionPerformed

    private void mIngDatActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_mIngDatActionPerformed
        // TODO add your handling code here:
        IngresoPacientesW ipw=new IngresoPacientesW();
        ipw.setVisible(true); // Hago visible la ventana de pacientes.
        dispose(); // Para que se oculte el menu.
    }//GEN-LAST:event_mIngDatActionPerformed

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(EnfermedadesMedW.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(EnfermedadesMedW.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(EnfermedadesMedW.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(EnfermedadesMedW.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>

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

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JMenuItem Salir;

```

```

private javax.swing.JButton anterior;
private javax.swing.JButton borrarC;
private javax.swing.JMenuItem cerrarSesion;
private javax.swing.JTextField codigoMed;
private javax.swing.JButton consultarBtn;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JMenu jMenu2;
private javax.swing.JMenu jMenu3;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JMenuItem jMenuItem2;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTextArea listaEnf;
private javax.swing.JMenuItem mINI;
private javax.swing.JMenuItem mInf;
private javax.swing.JMenuItem mIngDat;
private javax.swing.JTextField nombreMed;
private javax.swing.JMenu opciones;
// End of variables declaration//GEN-END:variables
}

```

#### ENCRYPTHELPER.JAVA

```

package controlpac;
import java.security.MessageDigest;
import java.util.Arrays;
import javax.crypto.Cipher;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
import org.apache.commons.codec.binary.Base64;

public class EncryptHelper {
    static String secretKey = "Pacientes"; //llave para encriptar datos

    public static String Encriptar(String texto) {
        String base64EncryptedString = "";
        try {
            MessageDigest md = MessageDigest.getInstance("MD5"); //Crea un hash con la clave elegida
            byte[] digestOfPassword = md.digest(secretKey.getBytes("utf-8"));
            byte[] keyBytes = Arrays.copyOf(digestOfPassword, 24);
            SecretKey key = new SecretKeySpec(keyBytes, "DESede"); //Crea la clave
            Cipher cipher = Cipher.getInstance("DESede");
            cipher.init(Cipher.ENCRYPT_MODE, key); //Inicializa el cifrado con la clave
            byte[] plainTextBytes = texto.getBytes("utf-8");
            byte[] buf = cipher.doFinal(plainTextBytes); //Cifra el texto
            byte[] base64Bytes = Base64.encodeBase64(buf); //Encodea el texto en base64
            base64EncryptedString = new String(base64Bytes);
        } catch (Exception ex) {
        }
    }
}

```

```

        return base64EncryptedString;
    }

    public static String Desencriptar(String textoEncriptado){
        String base64EncryptedString = "";
        try {
            byte[] message = Base64.decodeBase64(textoEncriptado.getBytes("utf-8")); //Desencodifica el texto en
base64
            MessageDigest md = MessageDigest.getInstance("MD5");
            byte[] digestOfPassword = md.digest(secretKey.getBytes("utf-8")); //Crea un hash con la clave elegida
            byte[] keyBytes = Arrays.copyOf(digestOfPassword, 24);
            SecretKey key = new SecretKeySpec(keyBytes, "DESede"); //Crea la clave
            Cipher decipher = Cipher.getInstance("DESede");
            decipher.init(Cipher.DECRYPT_MODE, key); //Inicia el descifrado
            byte[] plainText = decipher.doFinal(message); //Descifra el texto
            base64EncryptedString = new String(plainText, "UTF-8");
        } catch (Exception ex) {
        }
        return base64EncryptedString;
    }
}

```

## MODELO DE CALIDAD Y ALGORITMO

### CARACTERÍSTICAS Y SUBCARACTERÍSTICAS UTILIZADAS

#### *1. Funcionabilidad*

- a. Seguridad de Acceso

#### *2. Eficiencia*

- a. Utilización de recursos
- b. Comportamiento frente al tiempo

#### *3. Fiabilidad*

- a. Tolerancia a fallos
- b. Capacidad de recuperación de errores

#### *4. Mantenibilidad*

- a. Capacidad del código de ser analizado
- b. Capacidad del código de ser cambiado
- c. Estabilidad

#### *5. Usabilidad*

- a. Capacidad de ser entendido
- b. Capacidad de ser operado
- c. Capacidad de ser atractivo para el usuario

#### *6. Portabilidad*

a. Adaptabilidad

b. Instalabilidad

### ALGORITMO DE CALIDAD

El algoritmo de calidad consiste en evaluar cada subcaracterística como Mala, Regular o Buena, dependiendo de los criterios que se mencionarán más adelante.

Puntaje final de Subcaracterística	
Mala	0
Regular	1
Buena	2

Las combinaciones de los resultados de la evaluación de las subcaracterísticas determinarán el puntaje que se obtendrá de la característica. El paso siguiente es promediar los puntajes anteriores para obtener un valor general de la característica. Cada una de ellas tendrá un peso que se utilizará en el cálculo del puntaje final.

Característica	Peso
Funcionalidad	1
Eficiencia	2
Fiabilidad	2
Mantenibilidad	2
Usabilidad	2
Portabilidad	1

Cada peso se multiplicará por el puntaje obtenido de la característica correspondiente. Luego los puntajes pesados se promediarán para obtener el puntaje final. El promedio general obtenido determinará el nivel de calidad que podría ser: MUY SATISFACTORIO, SATISFACTORIO o NO SATISFACTORIO.

Nivel de Calidad	Puntaje Final
Muy Satisfactorio	Más de 14

Satisfactorio	Entre 7 y 14
No Satisfactorio	Menos de 7

Si alguna de las características tiene un puntaje de 0, el nivel de calidad del producto será NO SATISFACTORIO.

## CRITERIOS DE EVALUACIÓN DE MÉTRICAS

### 1. *Funcionabilidad*

#### SEGURIDAD DE ACCESO

##### **Descripción**

*Capacidad del producto software para asegurar la integridad de los datos y la confidencialidad de estos.*

Características a medir:

- Encriptación de datos
- Inicio de sesión de usuarios

Evaluación

- **Mala [0]** No cumple con alguna característica.
- **Regular [1]** Cumple con 1 característica.
- **Buena [2]** Cumple con 2 características.

### 2. *Eficiencia*

#### UTILIZACIÓN DE RECURSOS

##### **Descripción**

*Se evaluará la eficiencia del producto software de acuerdo al porcentaje de uso de procesador que realice.*

Evaluación

- **Mala [41; 100]** 41% o más de uso de procesador.
- **Regular [11; 40]** 11% a 40% de uso de procesador.
- **Buena [0;10]** 10% o menos de uso de procesador.

#### COMPORTAMIENTO EN EL TIEMPO

##### **Descripción**

*Se evaluará el tiempo que está el producto software sin informarle al usuario del estado en que se encuentra la solicitud que realizó.*

#### Evaluación

- **Mala [ $\geq 5$ ]** El producto está 5 o más segundos sin informar al usuario del estado de la solicitud.
- **Regular [2;4]** El producto está entre 2 y 4 segundos sin informar al usuario del estado de la solicitud.
- **Buena [0;1]** El producto está menos de 1 segundo sin informar al usuario del estado de la solicitud.

### 3. Fiabilidad

#### TOLERANCIA A FALLOS

---

##### Descripción

*Es la capacidad del producto software de mantener la integridad de los datos cuando se producen fallas del sistema.*

##### Características a medir:

- Cuando sucede un error se protegen los datos procesados.
- Se realiza un log de actividades que el sistema estaba haciendo.

#### Evaluación

- **Mala [0]** No cumple con alguna característica.
- **Regular [1]** Cumple con 1 característica.
- **Buena [2]** Cumple con 2 características.

#### CAPACIDAD DE RECUPERACIÓN DE ERRORES

---

##### Descripción

*Es la capacidad del sistema de reanudar sus actividades cuando se producen errores críticos.*

##### Características a medir:

- El sistema reanuda las actividades si se produce una falla crítica.
- Reanuda sus actividades y vuelve al estado en que estaba.

#### Evaluación

- **Mala [0]** No cumple con ninguna característica.
- **Regular [1]** Cumple con 1 característica.
- **Buena [2]** Cumple con 2 características.

### 4. Mantenibilidad

## CAPACIDAD DEL CÓDIGO PARA SER ANALIZADO

---

### Descripción

*Para evaluar la capacidad que tiene el código para ser analizado se tiene en cuenta el porcentaje de comentarios que posee el código por cada método y en general.*

### Evaluación

- **Mala [0; 14]** 14% o menos del código comentado.
- **Regular [15; 29]** Entre 15 y 29% del código comentado.
- **Buena [ $\geq 30$ ]** 30% o más del código comentado.

## CAPACIDAD DEL CODIGO PARA SER CAMBIADO

---

### Descripción

*Para evaluar la capacidad que tiene el código para ser cambiado se tomará en cuenta la complejidad ciclomática del método.*

### Evaluación

- **Mala [21]** La complejidad ciclomática es mayor o igual a 21.
- **Regular [11; 20]** La complejidad ciclomática es entre 11 y 20.
- **Buena [1; 10]** La complejidad ciclomática es menor o igual a 10.

## ESTABILIDAD

---

### Descripción

*Para determinar la estabilidad del software se evalúa el promedio de fallas que presenta el producto por prueba.*

### Evaluación

- **Mala [5]** El software presenta un promedio 5 o más errores por prueba.
- **Regular [2; 4]** El software presenta un promedio entre 2 y 4 errores por prueba.
- **Buena [0; 1]** El software presenta un promedio entre 0 y 1 error por prueba.

## 5. Usabilidad

## CAPACIDAD DE SER ENTENDIDO

---

### Descripción

*Capacidad que posee el software, para ayudar a los usuarios ante una determinada situación donde se necesite asistencia.*

#### Características a medir:

- Posee ayuda contextual sobre menús y botones de acción.
- Manual de usuario incorporado al sistema como un menú dedicado.

#### Evaluación

- **Mala [0]** No cumple con alguna característica.
- **Regular [1]** Cumple con 1 característica.
- **Buena [2]** Cumple con 2 características.

### CAPACIDAD PARA SER OPERADO

---

#### **Descripción**

*Es la Capacidad del producto software de ser utilizado sin asistencia adicional. Se valúa qué requiere el usuario para operar correctamente el producto.*

#### Evaluación

- **Mala [1]** El usuario requiere consultar a personal especializado para operar el producto software.
- **Regular [2]** El usuario requiere ayuda contextual y manual de uso para operar el producto software.
- **Buena [3]** El usuario opera el producto software sin asistencia.

### CAPACIDAD DE SER ATRACTIVO PARA EL USUARIO

---

#### **Descripción**

*Es la agrupación correcta de funcionalidad del producto software en su interfaz gráfica, desde su agrupación lógica hasta el número promedio de pasos para alcanzar una función o contenido específico.*

#### Evaluación

- **Mala [6]** 6 o más pasos promedio sin organización de categoría.
- **Regular [3; 5]** Entre 3 y 5 pasos promedio y distribuidos en categorías.
- **Buena [1; 2]** 1 o 2 pasos promedio y distribuidos en categorías.

## 6. Portabilidad

### ADAPTABILIDAD

---

#### **Descripción**

*Es la capacidad del producto software de adaptarse a diferentes sistemas operativos sin cambiar su estructura interna.*

#### Evaluación



- **Mala [1]** Compatible con 1 sistema operativo.
- **Regular [2]** Compatible con 2 sistemas operativos.
- **Buena [ $\geq 3$ ]** Compatible con 3 o más sistemas operativos.

## INSTALABILIDAD

---

### Descripción

*El producto software debe poder ser instalado en una cantidad mínima de pasos.*

### Evaluación

- **Mala [ $>7$ ]** El producto se instala en 7 o más pasos.
- **Regular [4; 6]** El producto se instala entre 4 y 6 pasos.
- **Buena [1; 3]** El producto se instala en 3 o menos pasos.

## MÉTODOS DE CAJA BLANCA

### MÉTODO A ANALIZAR

```
public int validarDatosPac(){
    String mensaje="";
    int ret=0; // comienza siendo valido.

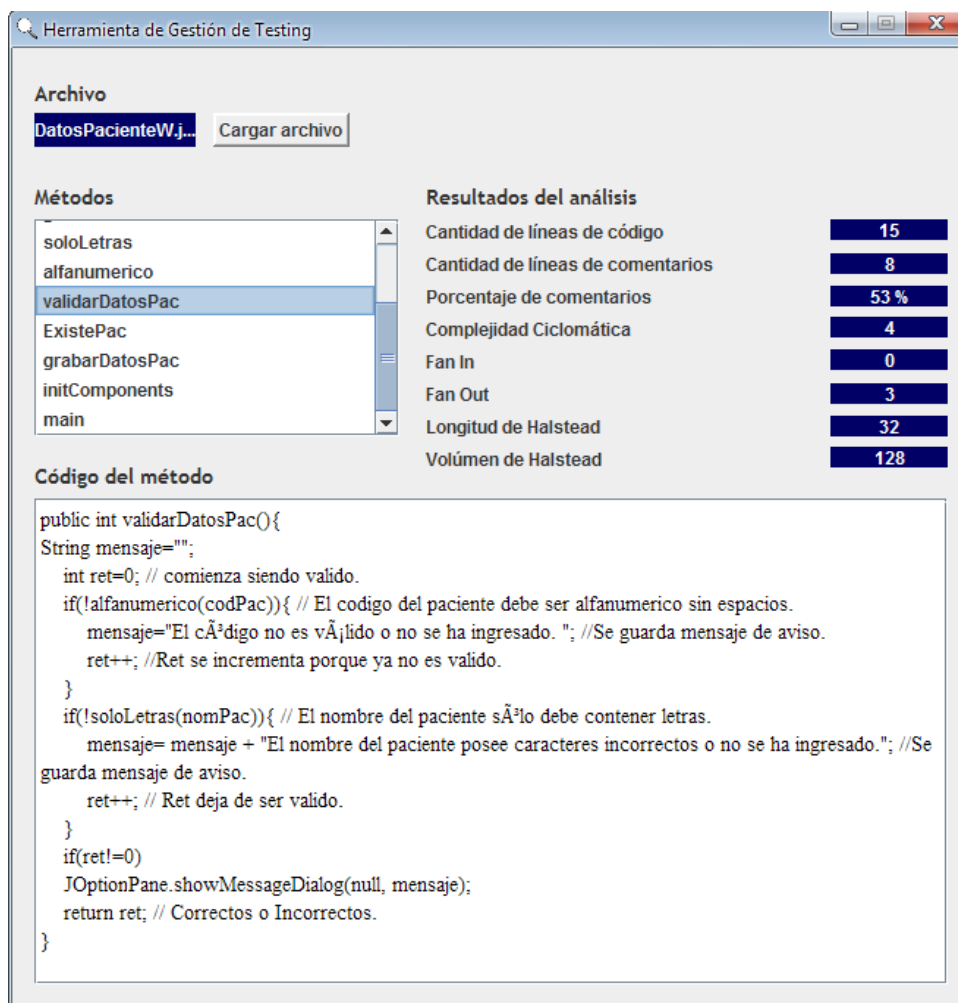
    if(!alfanumerico(codPac)){ // El codigo del paciente debe ser alfanumerico sin espacios.
        mensaje="El código no es válido o no se ha ingresado. "; //Se guarda mensaje de aviso.
        ret++; //Ret se incrementa porque ya no es valido.
    }

    if(!soloLetras(nomPac)){ // El nombre del paciente sólo debe contener letras.
        mensaje= mensaje + "El nombre del paciente posee caracteres incorrectos o no se ha ingresado.";
        //Se guarda mensaje de aviso.
        ret++; // Ret deja de ser valido.
    }

    if(ret!=0)
        JOptionPane.showMessageDialog(null, mensaje);

    return ret; // Correctos o Incorrectos.
}
```

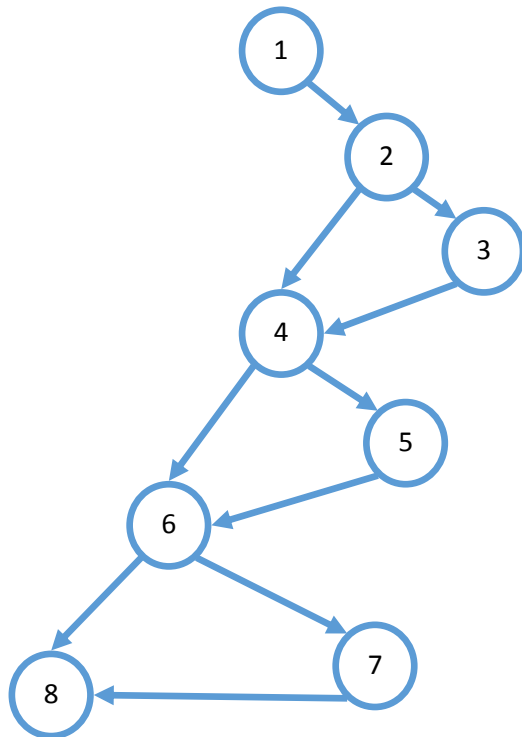
## REPORTE DE LA HERRAMIENTA DE TESTING



## MÉTODO DE MCCABE

```
public int validarDatosPac() {
1 | String mensaje="";
2 | int ret=0; // comienza siendo valido.
3 | if(!alfanumerico(codPac)){ // El codigo del paciente debe ser alfanumerico sin espacios.
4 |     mensaje="El código no es válido o no se ha ingresado. "; //Se guarda mensaje de aviso.
5 |     ret++; //Ret se incrementa porque ya no es valido.
6 | }
7 | if(!soloLetras(nomPac)){ // El nombre del paciente sólo debe contener letras.
8 |     mensaje= mensaje + "El nombre del paciente posee caracteres incorrectos o no se ha ingresado."; //Se
9 |     ret++; // Ret deja de ser valido.
10 | }
11 | if(ret!=0)
12 |     JOptionPane.showMessageDialog(null, mensaje);
13 | return ret; // Correctos o Incorrectos.
}
```

## Grafo



## Cálculo de la complejidad

Número de aristas del grafo:  $E = 10$

Número de nodos del grafo:  $V = 8$

$V(G) : E - V + 2 = 10 - 8 + 2 = 4$

## Caminos Linealmente Independientes

- 1) 1-2-4-6-8
- 2) 1-2-3-4-6-8 (No puede probarse)
- 3) 1-2-4-5-6-8 (No puede probarse)
- 4) 1-2-4-6-7-8 (No puede probarse)

## Casos de Prueba

Camino a testear	Nº Caso	<u>Datos de Entrada</u>	
		codpac	nompac
1)	1	P001	Juan Cruz

Los caminos 2) y 3) no pueden probarse ya que es inevitable que se deba pasar por el nodo 7 si se ha visitado el nodo 3 o 5, y estos caminos no lo consideran. El camino 4) tampoco puede ser probado ya

que, por el contrario, considera al nodo 7, pero es imposible pasar por él al no haber visitado previamente el nodo 3 y/o 5.

### MÉTODOS DE CAJA NEGRA

The screenshot shows a graphical user interface for data entry. The window title is 'Ingreso de Datos'. At the top, there is a menu bar with 'Ayuda', 'Ir a', and 'Opciones'. The main area is titled 'DATOS DEL PACIENTE'. It contains two input fields: 'Código:' and 'Nombre:'. Below these fields are three buttons: 'Borrar Campos', 'Ingresar', and 'Anterior'.

#### Identificación de las condiciones de entrada

- 1) **Tamaño Código de Paciente:** al menos 1 caracter.
- 2) **Tamaño Nombre de Paciente:** al menos 1 caracter.
- 3) **Tipo de dato Código de Paciente:** alfanumérico sin espacios.
- 4) **Tipo de dato Nombre de Paciente:** sólo letras y puede haber espacios pero no al comienzo.
- 5) **Primer carácter de Nombre de Paciente:** es una letra.

#### Clases de equivalencia

Condiciones de entrada	Clases				Regla que aplica
	Nro	Válidas	Nro	No Válidas	
Tamaño Código de Paciente	1	más de 1 carácter	6	Ningún caracter	Booleana
Tamaño Nombre de Paciente	2	más de 1 carácter	7	Ningún caracter	Booleana
Tipo de dato Código de Paciente	3	alfanumérico (sin espacios)	8	alfanumérico con espacios	Booleana
			9	no alfanuméricos	
Tipo de dato Nombre de Paciente	4	sólo letras y espacios	10	distinto de letras y espacios	Booleana

Primer caracter Nombre de Paciente	<b>5</b>	letra	<b>11</b>	distinto de letra	Booleana
--	----------	-------	-----------	-------------------	----------

## Casos de Prueba

### Clases Válidas

Nº Caso	Entrada			Clases que cubre
	Código de Paciente	Nombre de Paciente	op	
1	P001	Juan Cruz	N	<b>1,2,3,4,5</b>

### Clases No Válidas

Nº Caso	Entrada			Clases que cubre	Salida Esperada	Salida
	Código de Paciente	Nombre de Paciente	op			
2		-	-	<b>6</b>	Error: código vacío.	Error: no es un código válido. Vuelva a ingresar.
3	P002		-	<b>7</b>	Error: nombre vacío.	Error: nombre del paciente posee caracteres incorrectos o no se ha ingresado. Por favor, vuelva a ingresar un nuevo nombre.
4	P003 01	-	-	<b>8</b>	Error: código no válido.	Error: no es un código válido. Vuelva a ingresar.
5	@	-	-	<b>9</b>	Error: código no válido.	Error: no es un código válido. Vuelva a ingresar.
6	P003	Susan@	-	<b>10</b>	Error: nombre no es válido.	Error: nombre del paciente posee caracteres incorrectos o no se ha ingresado. Por favor, vuelva a ingresar un nuevo nombre.
7	P004	Şantiago	-	<b>11</b>	Error: nombre no es válido.	Error: nombre del paciente posee caracteres incorrectos o no se ha ingresado. Por favor, vuelva a ingresar un nuevo nombre.

## CUBIERTA DE CD




 Centro Médico Los Laureles	<div data-bbox="681 1160 1125 1321"><h3>Centro Médico Los Laureles</h3></div> <p data-bbox="347 1350 1358 1442">El programa Centro Médico Los Laureles brinda la funcionalidad de la administración de médicos y pacientes. Usted podrá llevar un registro de los mismos, así como obtener informes sobre enfermedades y pacientes atendidos.</p> <p data-bbox="381 1500 826 1529"><i>Requisitos mínimos de su computadora:</i></p> <ul data-bbox="381 1532 940 1686" style="list-style-type: none"><li>Al menos 124 MB de espacio libre en disco.</li><li>Java Runtime Environment versión 6 o superior.</li><li>Windows® Xp en adelante.</li><li>128 MB de memoria RAM.</li><li>Procesador Pentium 2 a 266 MHz.</li></ul> <p data-bbox="381 1722 699 1751"><i>El paquete software incluye:</i></p> <ul data-bbox="381 1753 1024 1814" style="list-style-type: none"><li>CD-ROM con el programa Centro Médico Los Laureles.</li><li>Manual de Usuario.</li></ul> <div data-bbox="359 1881 758 1926"> Universidad Nacional de la Matanza</div> <div data-bbox="1109 1686 1321 1892"></div> <p data-bbox="1109 1899 1332 1917">Grupo 14 - Hecho en Argentina</p>	Centro Médico Los Laureles 
	<p data-bbox="347 1960 1273 1982">Prohibida su reproducción total o parcial por cualquier medio digital y/o electrónico. Copyright © 2015, Grupo 14.</p>	



Centro Médico  
*Los Laureles*



 Universidad Nacional de la Matanza

Centro Médico  
*Los Laureles*



## PRUEBAS DE SISTEMA

### Entorno de Pruebas:

- Procesador: Intel i5 3200.
- Memoria RAM: 8 Gb.
- Disco Rígido: 500 Gb.
- Sistema Operativo: Windows 8.1 Professional

### PRUEBAS DE RECUPERACIÓN.

- 1) Mientras el usuario se encuentra registrando un nuevo paciente, se procederá a borrar el archivo de pacientes (datopac.txt) y verificar que el sistema puede continuar con el registro del mismo creando el archivo nuevamente e insertando el registro correspondiente.

*Su pudo observar que el sistema completó correctamente la prueba recreando el archivo borrado e insertando el registro del ingreso.*

### PRUEBAS DE SEGURIDAD.

- 1) Se intentara ingresar al sistema proveyendo credenciales no válidas y se verificará que el mismo deniegue el acceso.

*Se pudo observar que intentando ingresar al sistema con distintas credenciales no válidas detalladas a continuación el sistema denegó el acceso indicando con un mensaje la invalidez de las mismas:*

Usuario	Contraseña
admin	Prueba
admin	aaaaaaaaaaaaaaaaaaaa
admin	Admin
Admin	admin
aaaaaaa	admin
Prueba	Prueba

- 2) Se intentará ingresar al sistema proveyendo credenciales válidas verificando que el mismo permita el acceso.

*Se pudo observar que intentando ingresar al sistema con las credenciales válidas detalladas a continuación el sistema permitió el acceso:*

Usuario	Contraseña
admin	admin

- 3) Se abrirán los archivos mantenidos por el sistema (datopac.txt, datomed.txt y situpac.txt) y se intentará interpretar la información contenida por los mismos. Se deberá verificar que la misma se encuentre encriptada.



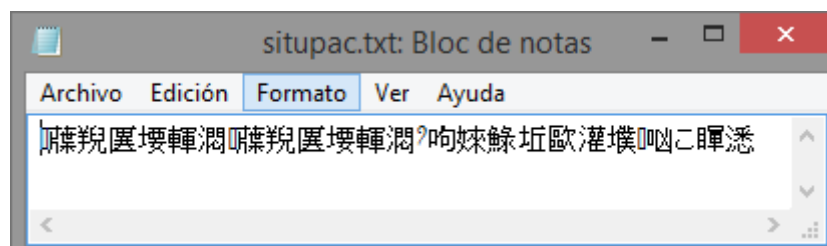
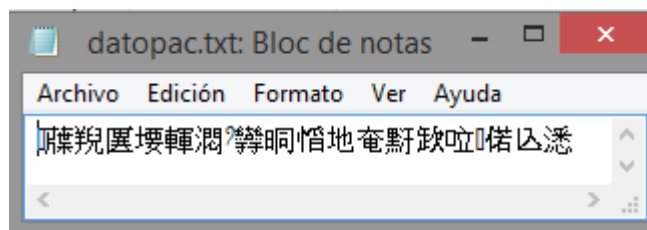
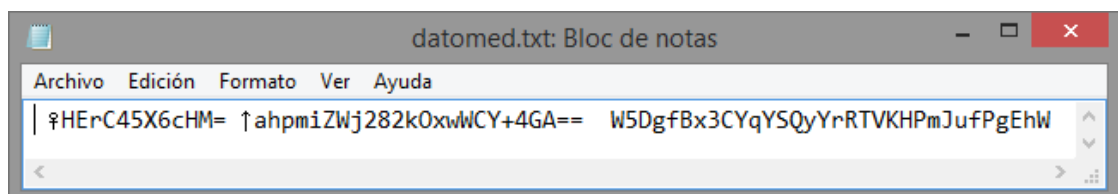
Se procedió a cargar a través del sistema datos en los tres archivos de la siguiente forma:

Registro de Pacientes (datopac.txt)	
Código	Nombre
1	Paciente Uno

Registro de Médicos (datomed.txt)		
Código	Nombre	Especialidad
1	Medico Uno	Especialidad Uno

Situación del Paciente (situpac.txt)		
Código del médico	Código del paciente	Diagnóstico
1	1	Diagnostico Uno

A abrir los mismos se pudo comprobar que la información no era legible tal como se puede observar en las siguientes capturas de pantalla:



## PRUEBAS DE RESISTENCIA.

- 1) Para comprobar si el programa puede manejar archivos de gran tamaño sin sufrir problemas de falta de memoria se generara un archivo de pacientes (datopac.txt) de forma automática con 10000 registros aleatorios y se verificara que el programa ejecute las operaciones de agregar paciente y consultar paciente correctamente.

Luego de generar el archivo se pudo comprobar que el programa responde a las operaciones correctamente.

## PRUEBAS DE RENDIMIENTO.

- 1) Se generara un archivo de médicos (datomed.txt) de forma automática con 10000 registros aleatorios y se verificara que el programa ejecute las operaciones de agregar médico y consultar médico en un tiempo no mayor a 2 segundos.

*Luego de generar el archivo se pudo comprobar que el programa responde a las operaciones correctamente y en un tiempo aproximado de 1 segundo.*

## ENCUESTA DE SATISFACCIÓN DE USUARIO

### FUNCIONALIDAD

1. ¿Fue capaz de ingresar al sistema sin conocer las credenciales de ingreso?:  
A. ☐ No. [2]    B. ☐ Si, luego de muchos intentos. [1]    C. ☐ Si, sin problemas. [0]
2. ¿Fue capaz de ingresar al sistema conociendo las credenciales de ingreso?  
A. ☐ Si, rápidamente. [2]    B. ☐ Sí, pero tuve problemas. [1]    C. ☐ No. [0]
3. ¿El software realiza correctamente las funciones que ofrece?  
A. ☐ Siempre. [2]    B. ☐ A veces. [1]    C. ☐ Nunca. [0]

### EFICIENCIA

4. ¿El programa se ejecuta lentamente?  
A. ☐ Sí, es demasiado lento. [0]    B. ☐ Su velocidad es normal. [1]    C. ☐ Es muy rápido. [2]
5. ¿Ha notado que su computadora funciona más lenta cuando se ejecuta el programa?  
A. ☐ Sí. [0]    B. ☐ No. [2]
6. ¿Con que velocidad obtuvo las respuestas cuando realizo consultas?  
A. ☐ Rápidamente. [2]    B. ☐ Su velocidad es normal. [1]    C. ☐ Demoró demasiado. [0]

### FIABILIDAD

7. ¿Pudo continuar utilizando el programa cuando éste informo algún error?  
A. ☐ Nunca. [0]    B. ☐ Casi nunca. [1]    C. ☐ Siempre. [2]
8. ¿Pudo continuar la tarea que estaba realizando luego de que el programa se detuviera inesperadamente?  
A. ☐ Nunca. [0]    B. ☐ Casi nunca. [1]    C. ☐ Siempre. [2]

9. ¿Los datos que ingreso en el programa se perdieron luego que el programa se detuviera inesperadamente?

- A. ☐ Todos. [0]      B. ☐ Algunos. [1]      C. ☐ Ninguno. [2]

#### MANTENIBILIDAD

10. ¿El programa se detuvo inesperadamente alguna vez durante su uso?

- A. ☐ Nunca. [2]      B. ☐ Casi nunca. [1]      C. ☐ Es muy frecuente esta situación. [0]

#### USABILIDAD

11. Utilizar el programa le resulta:

- A. ☐ Muy complicado. [0]      B. ☐ Fácil, pero a veces recorro al manual. [1]      C. ☐ Muy fácil. [2]

12. La manera en que presenta la información en pantalla es:

- A. ☐ Clara y entendible. [2]      B. ☐ Aceptable [1]      C. ☐ No me gusta su disposición. [0]

13. ¿Cómo calificaría la ayuda que brinda el sistema?

- A. ☐ Muy Buena [2]      B. ☐ Buena [1]      C. ☐ Regular [0]

#### PORTABILIDAD

14. ¿Tuvo algún inconveniente a la hora de instalar el programa?

- A. ☐ No [2]      B. ☐ Sí, pero pude solucionarlo. [1]      ☐ No lo pude instalar. [0]

15. ¿Pudo realizar la instalación rápidamente?

- A. ☐ Sí, fue muy simple y rápida. [2]      B. ☐ Sí, pero no fue rápida. [1]      C. ☐ No, fueron demasiados pasos. [0]

16. ¿Ha instalado el programa en equipos con diferentes sistemas operativos?

- A. ☐ Sí, sin problemas. [2]      B. ☐ Sí, pero no funciona correctamente. [1]      C. ☐ No. [0]

#### ALGORITMO PARA OBTENER EL PUNTAJE FINAL

Cada respuesta de la encuesta tiene un valor 2, 1 o 0 especificado entre corchetes. Los puntajes de las respuestas a las preguntas de cada característica evaluada se promediarán, obteniéndose un puntaje entre 0 y 2 para cada una de ellas.

Cada característica se ponderará según la siguiente tabla:

Característica	Peso
----------------	------

<b>Funcionalidad</b>	1
<b>Eficiencia</b>	2
<b>Fiabilidad</b>	2
<b>Mantenibilidad</b>	2
<b>Usabilidad</b>	2
<b>Portabilidad</b>	1

Cada peso se multiplicará por el puntaje obtenido de la característica correspondiente. Luego los puntajes pesados se promediarán para obtener el puntaje final. El promedio general obtenido determinará el nivel de calidad que podría ser: MUY SATISFACTORIO, SATISFACTORIO o NO SATISFACTORIO.

<b>Nivel de Calidad</b>	<b>Puntaje Final</b>
<b>Muy Satisfactorio</b>	Más de 14
<b>Satisfactorio</b>	Entre 7 y 14
<b>No Satisfactorio</b>	Menos de 7

Si alguna de las características tiene un puntaje de 0, el nivel de calidad del producto será NO SATISFACTORIO.

## RESULTADOS DE LA ENCUESTA

### PERFIL DE LOS ENCUESTADOS

**Encuestado Nº 1:** Hombre de 20 años, conocimientos intermedios de informática.

**Encuestado Nº 2:** Hombre de 55 años, conocimientos mínimos de informática.

**Encuestado Nº 3:** Mujer de 25 años, conocimientos avanzados de informática.

**Encuestado Nº 4:** Hombre de 50 años, conocimientos mínimos de informática.

**Encuestado Nº 5:** Mujer de 50 años, conocimientos intermedios de informática.

### RESPUESTAS OBTENIDAS

Característica	Pregunta	Encuestado Nº				
		1	2	3	4	5

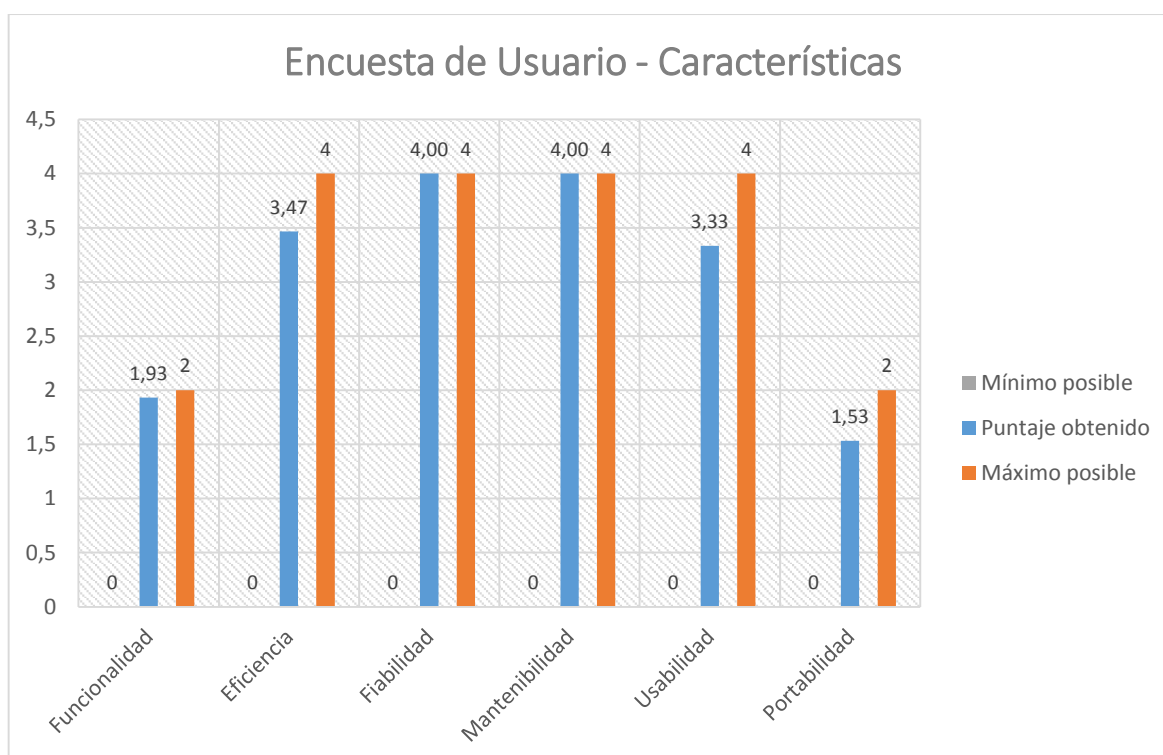
Funcionalidad	¿Fue capaz de ingresar al sistema sin conocer las credenciales de ingreso?	2	2	2	2	2
	¿Fue capaz de ingresar al sistema conociendo las credenciales de ingreso?	2	2	1	2	2
	¿El software realiza correctamente las funciones que ofrece?	2	2	2	2	2
Eficiencia	¿El programa se ejecuta lentamente?	1	2	1	2	2
	¿Ha notado que su computadora funciona más lenta cuando se ejecuta el programa?	2	2	2	2	2
	¿Con que velocidad obtuvo las respuestas cuando realizo consultas?	1	2	1	2	2
Fiabilidad	¿Pudo continuar utilizando el programa cuando éste informo algún error?	2	2	2	2	2
	¿Pudo continuar la tarea que estaba realizando luego de que el programa se detuviera inesperadamente?	2	2	2	2	2
	¿Los datos que ingreso en el programa se perdieron luego que el programa se detuviera inesperadamente?	2	2	2	2	2
Mantenibilidad	¿El programa se detuvo inesperadamente alguna vez durante su uso?	2	2	2	2	2
	¿Ha experimentado un problema luego de la actualización del software?	2	2	2	2	2
	¿Luego de una actualización ha detectado mal funcionamiento de alguna otra aplicación?	2	2	2	2	2
Usabilidad	Utilizar el programa le resulta	2	1	2	1	2
	La manera en que presenta la información en pantalla es:	1	2	2	2	1
	¿Cómo calificaría la ayuda que brinda el sistema?	1	2	2	2	2
Portabilidad	¿Tuvo algún inconveniente a la hora de instalar el programa?	2	1	2	2	1
	¿Pudo realizar la instalación rápidamente?	2	1	2	2	2
	¿Ha instalado el programa en equipos con diferentes sistemas operativos?	2	2	2	0	0

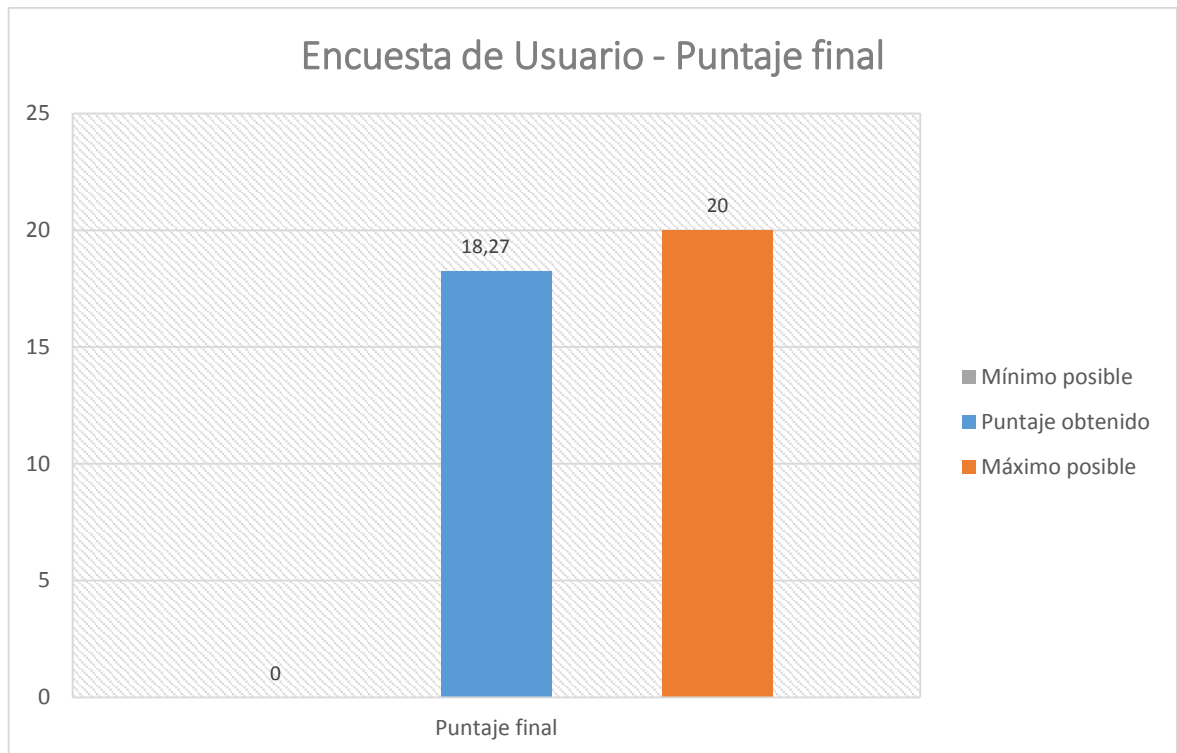
## CALIFICACIÓN OBTENIDA

Característica	Puntaje obtenido	Peso	Puntaje ponderado
Funcionalidad	1,93	1,00	1,93
Eficiencia	1,73	2,00	3,47
Fiabilidad	2,00	2,00	4,00
Mantenibilidad	2,00	2,00	4,00
Usabilidad	1,67	2,00	3,33
Portabilidad	1,53	1,00	1,53
<b>Puntaje final:</b>			<b>18,27</b>

La calificación final es: Muy Satisfactorio.

### GRÁFICO REPRESENTATIVO





## RESULTADOS DE EVALUACIÓN DE MÉTRICAS

### *1. Funcionabilidad*

#### SEGURIDAD DE ACCESO

##### Resultado:

Buena. El sistema encripta los datos de los archivos mantenidos e implementa una pantalla de inicio de sesión.

### *2. Eficiencia*

#### UTILIZACIÓN DE RECURSOS

##### Resultado:

Buena. Se pudo observar que el sistema mantiene un nivel de uso del procesador menor al 10 %.

#### COMPORTAMIENTO EN EL TIEMPO

##### Resultado:

Buena. Se pudo observar que el sistema mantiene tiempos de respuesta por debajo de los 2 segundos.

### *3. Fiabilidad*

#### TOLERANCIA A FALLOS

---

Resultado:

Regular. El sistema mantiene los archivos consistentes a pesar de haber ocurrido alguna falla, pero no mantiene un log de actividades.

#### CAPACIDAD DE RECUPERACIÓN DE ERRORES

---

Resultado:

Buena. Al producirse una falla el sistema reanuda las actividades y los archivos mantienen el estado en el que se encontraban.

### *4. Mantenibilidad*

#### CAPACIDAD DEL CÓDIGO PARA SER ANALIZADO

---

Resultado:

Buena. Se pudo observar en el método representativo (validarDatosPac de la clase DatosPacienteW un porcentaje de comentarios del 53 %.

#### CAPACIDAD DEL CODIGO PARA SER CAMBIADO

---

Resultado:

Buena. Se pudo observar en el método representativo (validarDatosPac de la clase DatosPacienteW una complejidad ciclomática de 4.

#### ESTABILIDAD

---

Resultado:

Buena. El sistema presenta un promedio de errores por prueba de 0.

### *5. Usabilidad*



## CAPACIDAD DE SER ENTENDIDO

---

### Resultado:

Buena. El sistema posee ayuda contextual sobre menús y botones de acción y anual de usuario incorporado al sistema como un menú dedicado.

## CAPACIDAD PARA SER OPERADO

---

### Resultado:

Buena. En las pruebas realizadas con los usuarios se pudo observar que utilizaron el software sin necesidad de asistencia.

## CAPACIDAD DE SER ATRACTIVO PARA EL USUARIO

---

### Resultado:

Buena. El sistema permite alcanzar cada una de sus funcionalidades en 2 pasos, estando las mismas agrupadas en categorías.

## *6. Portabilidad*

## ADAPTABILIDAD

---

### Resultado:

Buena. Se verificó que el sistema funciona en los sistemas operativos Windows XP SP3, Windows 7, Windows 8 y Windows 8.1.

## INSTALABILIDAD

---

### Resultado:

Regular. El sistema se puede instalar en 4 pasos.

## PUNTAJE OBTENIDO MEDIANTE EL ALGORITMO DE CALIDAD



La calificación final es: Muy Satisfactorio.

## COMPARACIÓN DE ENCUESTA DE USUARIO CON MODELO DE CALIDAD

