

Código Fuente



U.N.La.R

Si.G.A

Autores:

Lartigue Alejandro Gabriel y

Luques Jorge Matías

2023

Índice de contenido

Paquete Clases	1
ConexionBD.Java	1
GenerarCodigos.Java	2
Logo.Java	3
Main.Java	4
MySQL.Java	5
PantallaPrincipal.Java	8
Paquete Consultas	12
BuscarAlumno.Java	12
BuscarAlumnoDerivacion.Java	14
BuscarAlumnoDerivacionG.Java	16
BuscarAlumnoG.Java	18
BuscarAlumnoRetorno.Java	20
BuscarAlumnoRetornoG.Java	22
BuscarAnimalDerivacion.Java	24
BuscarAnimalDerivacionG.Java	26
BuscarCasoClinico.Java	28
BuscarCasoClinicoG.Java	29
BuscarCasoClinicoMostrarRetornoAG.Java	30
BuscarCasoClinicoMostrarRetornoAP.Java	31
BuscarDocente.Java	32
BuscarDocenteDerivacion.Java	34
BuscarDocenteDerivacionG.Java	36
BuscarDocenteG.Java	38
BuscarDocenteRetorno.Java	40
BuscarDocenteRetornoG.Java	42
BuscarDueñoDerivacion.Java	44
BuscarDueñoDerivacionG.Java	46
BuscarDueñoG.Java	48
BuscarDueñoP.Java	50
BuscarDueñoRetorno.Java	52
BuscarMostrarCasoClinico.Java	56

BuscarMostrarCascoClinicoG.Java.....	57
ConsultaAnimalG.Java	58
ConsultaAnimalP.Java	63
DerivacionP.Java.....	76
DetalleP.Java	79
ListadoDerivacionG.Java.....	80
ListadoDerivacionP.Java	82
ListadoMostrarRetornoAG.Java.....	84
ListadoMostrarRetornoAP.Java	86
MostrarConsultaAnimalG.Java	88
MostrarConsultaAnimalP.Java.....	95
MostrarDerivacionAnimalG.Java	109
MostrarDerivacionAnimalP.Java.....	112
MostrarRetornoAnimalG.Java	115
MostrarRetornoAnimalP.Java.....	122
RetornoAnimalG.Java	129
RetornoAnimalP.Java	135
Paquetes Exámenes	141
BuscarCasoClinico.Java.....	141
BuscarDocente.Java.....	144
Hemograma.Java	146
QuimicaClinica.Java	149
Urianalisis.Java	151
Paquete ModuloAnimal	154
BuscarAnimalP.Java.....	154
DetalleAnimal.Java	156
EspecieRazaAnimalG.Java.....	157
EspecieRazaAnimalP.Java	162
ImgAnimalp.Java	167
ListaDueño.Java.....	168
ListaDueñoG.Java	170
ModificarAnimalG.Java.....	172
ModificarAnimalP.Java	176
NuevoAnimalG.Java.....	180

NuevoAnimalP.Java	183
Paquete ModuloDueño	186
ImgDuenio.Java	186
ModificarDueño.Java	187
NuevoDueño.Java	189
Paquetes ModuloUsuario	190
ImgDoctor.Java	190
ModificarAlumno.Java	191
ModificarDoctor.Java	193
NuevoAlumno.Java	195
NuevoDoctor.Java	196
Paquete Reportes	197
BuscarHemograma.Java	197
BuscarQuimicaClinica.Java	199
BuscarUrianalisys.Java	201
ListadoMostrarRetornoAP.Java	203
IniciarReporte.Java	207
IniciarReporte2.Java	209
IniciarReporte3.Java	211
ReporteHemograma.Java	213

Paquete Clases

ConexionBD.Java

```
package Clases;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
public class ConexionBD {
    static String bd = "siga";
    static String login = "root";
    static String password = "";
    static String url = "jdbc:mysql://localhost/" + bd;
    Connection conn = null;
    /** Constructor de DbConnection */
    public ConexionBD() {
        try{
            //obtenemos el driver de para mysql
            Class.forName("com.mysql.jdbc.Driver");
            //obtenemos la conexión
            conn = DriverManager.getConnection(url,login,password);
            if (conn!=null){
                System.out.println("Conección a base de datos "+bd+" listo");
            }
        }catch(SQLException e){
            System.out.println(e);
        }catch(ClassNotFoundException e){
            System.out.println(e);
        }
    }
    /**Permite retornar la conexión*/
    public Connection getConnection(){
        return conn;
    }
    public void desconectar(){
        conn = null;
        System.out.println("La conexion a la base de datos "+bd+" a terminado");
    }
}
```

GenerarCodigos.Java

```
package Clases;
public class GenerarCodigos {
    private int dato;
    private int cont=1;
    private String num="";
    public void generar(int dato) {
        this.dato = dato;
        if((this.dato>=1000) || (this.dato<10000)){
            int can=cont+this.dato;
            num = "" + can;
        }
        if((this.dato>=100) || (this.dato<1000)){
            int can=cont+this.dato;
            num = "0" + can;
        }
        if((this.dato>=9) || (this.dato<100)){
            int can=cont+this.dato;
            num = "00" + can;
        }
        if (this.dato<9){
            int can=cont+this.dato;
            num = "000" + can;
        }
    }
    public String serie(){
        return this.num;
    }
}
```

Logo.Java

```
package Clases;
import java.awt.Graphics;
import javax.swing.ImageIcon;
public class Logo extends javax.swing.JPanel {
    public Logo(){
        this.setSize(600,300);//se selecciona el tamaño del panel
    }
    @Override
    //Se crea un método cuyo parámetro debe ser un objeto Graphics
    public void paint(Graphics g){
        //se dibuja la imagen que tenemos en el paquete
        ImageIcon imagenFondo = new ImageIcon(getClass().getResource("/imagenes/logo.png"));
        g.drawImage(imagenFondo.getImage(),0,0,600,300, null);
        setOpaque(false);
        super.paintComponent(g);
    }
}
```

Main.Java

```
package Clases;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import org.jvnet.substance.SubstanceLookAndFeel;
public class Main {
    public static void main(String[] args) {
        //probamos la conexion a la base de datos
        ConexionBD test = new ConexionBD();
        //codigo substance
        JFrame.setDefaultLookAndFeelDecorated(true); //que nos permite dejar a Substance la decoracion ( por asi decirlo)
        SubstanceLookAndFeel.setSkin("org.jvnet.substance.skin.OfficeBlue2007Skin");//aplicamos skin deseado
        //SubstanceLookAndFeel.setCurrentTheme("org.jvnet.substance.theme.SubstanceDarkVioletTheme"); // Se aplica el tema Aqui de Substance
        SubstanceLookAndFeel.setCurrentWatermark("org.jvnet.substance.watermark.SubstanceGenericNoiseWatermark");//aplicamos marca de agua
        PantallaPrincipal principal = new PantallaPrincipal();
        principal.show();
    }
}
```


MySQL.Java

```
package Clases;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JOptionPane;
public class MySQL {
    ConexionBD con;
    public MySQL () {
        con = new ConexionBD();
    }
    public int contarregistros(String tabla){
        int registros=0;
        try{
            PreparedStatement pstmt = con.getConnection().prepareStatement("SELECT count(1) as total FROM " + tabla);
            ResultSet res = pstmt.executeQuery();
            res.next();
            registros = res.getInt("total");
            res.close();
        }catch(SQLException e){
            System.out.println(e);
        }
        return registros;
    }
    /* METODO PARA DETERMINAR SI UN REGISTRO YA EXISTE EN LA BASE DE DATOS TIENE COMO PARAMETROS (nombre de la tabla, nombre
    columna*/
    public boolean existe(String tabla, String colName, String id){
        int registros = 0;
        try{
            PreparedStatement pstmt = con.getConnection().prepareStatement("SELECT count(1) as total FROM " + tabla + " where " + colName + " = ?");
            pstmt.setString(1, id);
            ResultSet res = pstmt.executeQuery();
            res.next();
            registros = res.getInt("total");
            res.close();
        }catch(SQLException e){
            System.out.println(e);
        }
        if (registros > 0){return true;}else{return false;}
    }
    /* INSERTA UN NUEVO REGISTRO EN LA BASE DE DATOS PARAMETROS(Un array de String con los datos a insertar,la instruccion sql)*/
    public void InsertarRegistro(String datos[], String sql){
        try {
            PreparedStatement pstmt = con.getConnection().prepareStatement(sql);
            for(int i=0; i<=datos.length-1;i++){
                pstmt.setString(i+1, datos[i]);
            }
            int resp = JOptionPane.showConfirmDialog(null, "¿Esta seguro?", "Alerta!", JOptionPane.YES_NO_OPTION);
            if (resp==0){
                pstmt.execute();
                pstmt.close();
                JOptionPane.showMessageDialog(null, "Registro Guardado");
            }catch(SQLException e){
                System.out.println(e);
            }
        }
    }
    /* BORRA UN REGISTRO DE LA BASE DE DATOS parametros (Nombre de la tabla, nombre columna,Codigo unico)*/
    public void DeleteRegistro(String tabla, String col, String id){
        try {
            PreparedStatement pstmt = con.getConnection().prepareStatement("delete from " + tabla + " where " + col + " = ?");
            pstmt.setString(1, id);
            pstmt.execute();
            pstmt.close();
        }catch(SQLException e){
            System.out.println(e);
        }
    }
}
```

```
}
/* ACTUALIZA UNA TABLA DE LA BASE DE DATOS parametros (Un Array de string con los datos a actualizar, la instruccion sql )*/
public void UpdateRegistro(String datos[], String sql){
    try {
        PreparedStatement pstm = con.getConnection().prepareStatement(sql);
        for(int i=0; i<=datos.length-1;i++){
            pstm.setString(i+1, datos[i]);
        }
        int resp = JOptionPane.showConfirmDialog(null, "¿Esta seguro?", "Alerta!", JOptionPane.YES_NO_OPTION);
        if (resp==0){
            pstm.execute();
            pstm.close();
            JOptionPane.showMessageDialog(null, "Registro Actualizado");
        }
    }catch(SQLException e){
        System.out.println(e);
    }
}

/* METODO PARA OBTENER TODOS LOS DATOS DE UNA TABLA parametros (Un array con los nombres de las columnas, el nombre de la tabla, la instruccion sql)*/
public Object [][] GetTabla(String colName[], String tabla, String sql){
    int registros = 0;
    //obtenemos la cantidad de registros existentes en la tabla
    try{
        PreparedStatement pstm = con.getConnection().prepareStatement("SELECT count(1) as total FROM " + tabla);
        ResultSet res = pstm.executeQuery();
        res.next();
        registros = res.getInt("total");
        res.close();
    }catch(SQLException e){
        System.out.println(e);
    }
    Object[][] data = new String[registros][colName.length];
    String col[] = new String[colName.length];
    //realizamos la consulta sql y llenamos los datos en "Object"
    try{
        PreparedStatement pstm = con.getConnection().prepareStatement(sql);
        ResultSet res = pstm.executeQuery();
        int i = 0;
        while(res.next()){
            for(int j=0; j<=colName.length-1;j++){
                col[j] = res.getString(colName[j]);
                data[i][j] = col[j];
            }
            i++;
        }
        res.close();
    }catch(SQLException e){
        System.out.println(e);
    }
    return data;
}

/* METODO PARA TODOS LOS DATOS DE UNA COLUMNA parametros (Nombre de la tabla, nombre columna, instruccion sql )*/
public Object [] GetColumna(String tabla, String colName, String sql){
    int registros = 0;
    //obtenemos la cantidad de registros existentes en la tabla
    try{
        PreparedStatement pstm = con.getConnection().prepareStatement("SELECT count(1) as total FROM " + tabla);
        ResultSet res = pstm.executeQuery();
        res.next();
        registros = res.getInt("total");
        res.close();
    }catch(SQLException e){
        System.out.println(e);
    }
    Object[] data = new String[registros];
    try{
        PreparedStatement pstm = con.getConnection().prepareStatement(sql);
        ResultSet res = pstm.executeQuery();
        int i = 0;
        while(res.next()){
```

```
        data[i] = res.getString(colName);
        i++;
    }
    res.close();
} catch(SQLException e){
    System.out.println(e);
}
return data;
}
/* METODO PARA OBTENER UN DATO DE UNA TABLA PARAMETROS (nombre columna, instruccion sql)*/
public String GetData(String colName, String sql){
    String data = new String();
    try{
        PreparedStatement pstm = con.getConnection().prepareStatement(sql);
        ResultSet res = pstm.executeQuery();
        while(res.next()){
            data = res.getString(colName);
        }
        res.close();
    } catch(SQLException e){
        System.out.println(e);
    }
    return data;
}
public boolean existe2(String tabla, String colName, String colName2, String id, String id2, String id3){
    int registros = 0;
    try{
        PreparedStatement pstm = con.getConnection().prepareStatement("SELECT count(1) as total FROM " + tabla + " where " + colName + " = ? AND " + colName2 + " BETWEEN ? AND ?");
        pstm.setString(1, id);
        pstm.setString(2, id2);
        pstm.setString(3, id3);
        ResultSet res = pstm.executeQuery();
        res.next();
        registros = res.getInt("total");
        res.close();
    } catch(SQLException e){
        System.out.println(e);
    }
    if (registros > 0){return true;}else{return false;}
}
}
```

PantallaPrincipal.Java

```
package Clases;
import Consultas.BuscarCasoClinico;
import Consultas.BuscarCasoClinicoG;
import Consultas.BuscarCasoClinicoMostrarRetornoAG;
import Consultas.BuscarCasoClinicoMostrarRetornoAP;
import Consultas.BuscarMostrarCasoClinico;
import Consultas.BuscarMostrarCasoClinicoG;
import Consultas.ConsultaAnimalG;
import Consultas.ConsultaAnimalP;
import Consultas.DerivacionG;
import Consultas.DerivacionP;
import Consultas.ListadoDerivacionG;
import Consultas.ListadoDerivacionP;
import Consultas.RetornoAnimalP;
import Examenes.Hemograma;
import Examenes.QuimicaClinica;
import Examenes.Urianalisis;
import ModuloAnimal.BuscarAnimalP;
import ModuloAnimal.EspecieRazaAnimalG;
import ModuloAnimal.EspecieRazaAnimalP;
import ModuloAnimal.ModificarAnimalG;
import ModuloAnimal.ModificarAnimalP;
import ModuloAnimal.NuevoAnimalG;
import ModuloAnimal.NuevoAnimalP;
import ModuloDueño.ModificarDueño;
import ModuloDueño.NuevoDueño;
import ModuloUsuario.ModificarAlumno;
import ModuloUsuario.ModificarDoctor;
import ModuloUsuario.NuevoAlumno;
import ModuloUsuario.NuevoDoctor;
import Reportes.BuscarHemograma;
import Reportes.BuscarQuimicaClinica;
import Reportes.BuscarUrianalisis;
import Reportes.EstadisticaAnálisisDocenteAlumno;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import org.jvnet.substance.SubstanceLookAndFeel;

public class PantallaPrincipal extends javax.swing.JFrame {
    /*Creates new form PantallaPrincipal*/
    public PantallaPrincipal() {
        initComponents();
        this.setExtendedState(MAXIMIZED_BOTH);
        Logo img = new Logo ();
        jpLogo.add(img);
        jpLogo.repaint();
        try{
            setIconImage(new ImageIcon(getClass().getResource("../Imagenes/h.png")).getImage());
        }catch(Exception ex){
        }
    }
    /**
     * 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")
    private void jmiNuevoDuenioActionPerformed(java.awt.event.ActionEvent evt) {
        NuevoDueño duenio = new NuevoDueño();
        jdpFondo.add(duenio);
        duenio.show();
    }
    private void jmiSalirActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    private void jmiModificarDueñoActionPerformed(java.awt.event.ActionEvent evt) {
        ModificarDueño duenio = new ModificarDueño();
    }
}
```

```
jdpFondo.add(duenio);
duenio.show();
}
private void jmiNuevoAnimalPActionPerformed(java.awt.event.ActionEvent evt) {
    NuevoAnimalP duenio = new NuevoAnimalP();
    jdpFondo.add(duenio);
    duenio.show();
}
private void jmiModificarAnimalPActionPerformed(java.awt.event.ActionEvent evt) {
    ModificarAnimalP duenio = new ModificarAnimalP();
    jdpFondo.add(duenio);
    duenio.show();
}
private void jmiNuevoAnimalGActionPerformed(java.awt.event.ActionEvent evt) {
    NuevoAnimalG animal = new NuevoAnimalG();
    jdpFondo.add(animal);
    animal.show();
}
private void jmiModificarAnimalGActionPerformed(java.awt.event.ActionEvent evt) {
    ModificarAnimalG animal = new ModificarAnimalG();
    jdpFondo.add(animal);
    animal.show();
}
private void jmiNuevoDoctorActionPerformed(java.awt.event.ActionEvent evt) {
    NuevoDoctor doctor = new NuevoDoctor();
    jdpFondo.add(doctor);
    doctor.show();
}
private void jmiModificarDoctorActionPerformed(java.awt.event.ActionEvent evt) {
    ModificarDoctor doctor = new ModificarDoctor();
    jdpFondo.add(doctor);
    doctor.show();
}
private void jmiNuevoAlumnoActionPerformed(java.awt.event.ActionEvent evt) {
    NuevoAlumno alumno = new NuevoAlumno();
    jdpFondo.add(alumno);
    alumno.show();
}
private void jmiModificarAlumnoActionPerformed(java.awt.event.ActionEvent evt) {
    ModificarAlumno alumno = new ModificarAlumno();
    jdpFondo.add(alumno);
    alumno.show();
}
private void jmiNuevaRazaActionPerformed(java.awt.event.ActionEvent evt) {
    EspecieRazaAnimalP raza = new EspecieRazaAnimalP();
    jdpFondo.add(raza);
    raza.show();
}
private void jmiNuevaRazaGActionPerformed(java.awt.event.ActionEvent evt) {
    EspecieRazaAnimalG raza = new EspecieRazaAnimalG();
    jdpFondo.add(raza);
    raza.show();
}
private void jmiConsultaAnimalPActionPerformed(java.awt.event.ActionEvent evt) {
    ConsultaAnimalP raza = new ConsultaAnimalP();
    jdpFondo.add(raza);
    raza.show();
}
private void jmiConsultaAnimalGActionPerformed(java.awt.event.ActionEvent evt) {
    ConsultaAnimalG consulta = new ConsultaAnimalG();
    jdpFondo.add(consulta);
    consulta.show();
}
private void jmiRetornoAnimalPActionPerformed(java.awt.event.ActionEvent evt) {
    BuscarCasoClinico buscar = new BuscarCasoClinico();
    jdpFondo.add(buscar);
    buscar.show();
}
private void jmiRetornoAnimalGActionPerformed(java.awt.event.ActionEvent evt) {
    BuscarCasoClinicoG buscar = new BuscarCasoClinicoG();
    jdpFondo.add(buscar);
}
```

```
        buscar.show();
    }
    private void jmiHemogramaActionPerformed(java.awt.event.ActionEvent evt) {
        Hemograma analisis = new Hemograma();
        jdpFondo.add (analisis);
        analisis.show();
    }
    private void jmiQuimicaActionPerformed(java.awt.event.ActionEvent evt) {
        QuimicaClinica analisis = new QuimicaClinica();
        jdpFondo.add (analisis);
        analisis.show();
    }
    private void jmiUrianálisisActionPerformed(java.awt.event.ActionEvent evt) {
        Urianálisis analisis = new Urianálisis();
        jdpFondo.add (analisis);
        analisis.show();
    }
    private void ReporteHemogramaActionPerformed(java.awt.event.ActionEvent evt) {
        BuscarHemograma reporte = new BuscarHemograma();
        jdpFondo.add (reporte);
        reporte.show();
    }
    private void ReporteQuimicaActionPerformed(java.awt.event.ActionEvent evt) {
        BuscarQuimicaClinica reporte = new BuscarQuimicaClinica();
        jdpFondo.add (reporte);
        reporte.show();
    }
    private void UrianálisisActionPerformed(java.awt.event.ActionEvent evt) {
        BuscarUrianálisis reporte = new BuscarUrianálisis();
        jdpFondo.add (reporte);
        reporte.show();
    }
    private void Consulta1ActionPerformed(java.awt.event.ActionEvent evt) {
        EstadisticaAnálisisDocenteAlumno reporte = new EstadisticaAnálisisDocenteAlumno();
        jdpFondo.add (reporte);
        reporte.show();
    }
    private void jmiDerivacionPActionPerformed(java.awt.event.ActionEvent evt) {
        DerivacionP derivacion = new DerivacionP ();
        jdpFondo.add (derivacion);
        derivacion.show();
    }
    private void jmiListaDerivacionesPActionPerformed(java.awt.event.ActionEvent evt) {
        ListadoDerivacionP derivacion = new ListadoDerivacionP ();
        jdpFondo.add (derivacion);
        derivacion.show();
    }
    private void jmiVerConsultaAnimalPActionPerformed(java.awt.event.ActionEvent evt) {
        BuscarMostrarCasoClinico consulta = new BuscarMostrarCasoClinico ();
        jdpFondo.add (consulta);
        consulta.show();
    }
    private void jmiListadoRetornoAnimalPActionPerformed(java.awt.event.ActionEvent evt) {
        BuscarCasoClinicoMostrarRetornoAP retorno = new BuscarCasoClinicoMostrarRetornoAP ();
        jdpFondo.add (retorno);
        retorno.show();
    }
    private void jmiVerConsultaAnimalGActionPerformed(java.awt.event.ActionEvent evt) {
        BuscarMostrarCasoClinicoG casoClinico = new BuscarMostrarCasoClinicoG();
        jdpFondo.add (casoClinico);
        casoClinico.show();
    }
    private void jmiListadoRetornoAnimalGActionPerformed(java.awt.event.ActionEvent evt) {
        BuscarCasoClinicoMostrarRetornoAG retorno = new BuscarCasoClinicoMostrarRetornoAG ();
        jdpFondo.add (retorno);
        retorno.show();
    }
    private void jmiListaDerivacionesGActionPerformed(java.awt.event.ActionEvent evt) {
        ListadoDerivacionG derivacion = new ListadoDerivacionG ();
        jdpFondo.add (derivacion);
        derivacion.show();
    }
```



```
}
private void jmiDerivacionGActionPerformed(java.awt.event.ActionEvent evt) {
    DerivacionG derivacion = new DerivacionG ();
    jdpFondo.add (derivacion);
    derivacion.show();
}
/* @param args the command line arguments */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new PantallaPrincipal().setVisible(true);
        }
    });
}
// Variables declaration - do not modify
private javax.swing.JMenuItem Consulta1;
private javax.swing.JMenu MenuEstadistica;
private javax.swing.JMenuItem ReporteHemograma;
private javax.swing.JMenuItem ReporteQuimica;
private javax.swing.JMenuItem Urianálisis;
private javax.swing.JMenu jMenu1;
private javax.swing.JMenu jMenuAlumno;
private javax.swing.JMenu jMenuAnálisis;
private javax.swing.JMenu jMenuAnimal;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JMenu jMenuConsulta;
private javax.swing.JMenu jMenuDerivacion;
private javax.swing.JMenu jMenuDoctor;
private javax.swing.JMenu jMenuDuenio;
private javax.swing.JMenu jMenuReportes;
private javax.swing.JMenu jMenuRetorno;
private javax.swing.JMenu jMenuSistema;
public static javax.swing.JDesktopPane jdpFondo;
private javax.swing.JMenuItem jmiConsultaAnimalG;
private javax.swing.JMenuItem jmiConsultaAnimalP;
private javax.swing.JMenuItem jmiDerivacionG;
private javax.swing.JMenuItem jmiDerivacionP;
private javax.swing.JMenuItem jmiHemograma;
private javax.swing.JMenuItem jmiListaDerivacionesG;
private javax.swing.JMenuItem jmiListaDerivacionesP;
private javax.swing.JMenuItem jmiListadoRetornoAnimalG;
private javax.swing.JMenuItem jmiListadoRetornoAnimalP;
private javax.swing.JMenuItem jmiModificarAlumno;
private javax.swing.JMenuItem jmiModificarAnimalG;
private javax.swing.JMenuItem jmiModificarAnimalP;
private javax.swing.JMenuItem jmiModificarDoctor;
private javax.swing.JMenuItem jmiModificarDueño;
private javax.swing.JMenuItem jmiNuevaRaza;
private javax.swing.JMenuItem jmiNuevaRazaG;
private javax.swing.JMenuItem jmiNuevoAlumno;
private javax.swing.JMenuItem jmiNuevoAnimalG;
private javax.swing.JMenuItem jmiNuevoAnimalP;
private javax.swing.JMenuItem jmiNuevoDoctor;
private javax.swing.JMenuItem jmiNuevoDuenio;
private javax.swing.JMenuItem jmiQuimica;
private javax.swing.JMenuItem jmiRetornoAnimalG;
private javax.swing.JMenuItem jmiRetornoAnimalP;
private javax.swing.JMenuItem jmiSalir;
private javax.swing.JMenuItem jmiUrianálisis;
private javax.swing.JMenuItem jmiVerConsultaAnimalG;
private javax.swing.JMenuItem jmiVerConsultaAnimalP;
private javax.swing.JPanel jpLogo;
// End of variables declaration
}
```

Paquete Consultas

BuscarAlumno.Java

```
package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import ModuloUsuario.ImgDoctor;
import javax.swing.ImageIcon;
import javax.swing.table.DefaultTableModel;
public class BuscarAlumno extends javax.swing.JInternalFrame {
    public BuscarAlumno() {
        initComponents();
        ImgDoctor img = new ImgDoctor();
        jplImagen.add(img);
        jplImagen.repaint();
        cargarTabla();
    }
    private void cargarTabla(){
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_alumno", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        dtAlumno = alumno.GetTabla(colName, "alumno", "SELECT * FROM `alumno` ORDER BY `apellido`");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtAlumno, columnNames);
        jtAlumno.setModel(datos);
    }
    /* 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")
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar= jtAlumno.getSelectedRow();
        if(filamodificar>=0){
            String apellido= jtAlumno.getValueAt(filamodificar, 0).toString();
            String nombre= jtAlumno.getValueAt(filamodificar, 1).toString();
            String dni= jtAlumno.getValueAt(filamodificar, 2).toString();
            ConsultaAnimalP.jlAlumno.setText(apellido);
            ConsultaAnimalP.jlAlumno1.setText(nombre);
            ConsultaAnimalP.jlAlumno2.setText(dni);
        }
        this.dispose();
    }
    private void jtAlumnoMouseClicked(java.awt.event.MouseEvent evt) {
        fila=jtAlumno.rowAtPoint(evt.getPoint());
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String texto= jtfBuscar.getText();
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_alumno", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        if (jrbDni.isSelected() == true) {
            dtAlumno = alumno.GetTabla(colName, "alumno", "SELECT * FROM `duenio` WHERE `dni_alumno`=" + texto + " ");
        }
        if (jrbApellido.isSelected() == true) {
            dtAlumno = alumno.GetTabla(colName, "alumno", "SELECT * FROM `doctor` WHERE `apellido` = \"\" + texto + \"\"");
        }
        DefaultTableModel datos = new DefaultTableModel(dtAlumno, columnNames);
        jtAlumno.setModel(datos); // se colocan los datos en la tabla
        //depurador de jtDuenio de item null
        int a = jtAlumno.getRowCount();
        int i=0;
        a--;
    }
}
```

```
while(a>=i){
    if(jtAlumno.getValueAt(i,1)==null){
        ((DefaultTableModel)jtAlumno.getModel()).removeRow(i);
        a--;
        i--;
    }
    i++;
}
}
Object[][] dtAlumno;
MySQL alumno = new MySQL();
int fila = -1;
// Variables declaration - do not modify
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JButton jbBuscar;
private javax.swing.JButton jbCancelar;
private javax.swing.JButton jbElegir;
private javax.swing.JPanel jPanelImagen;
private javax.swing.JRadioButton jrbApellido;
private javax.swing.JRadioButton jrbDni;
public static javax.swing.JTable jtAlumno;
private javax.swing.JTextField jtfBuscar;
// End of variables declaration
}
```

BuscarAlumnoDerivacion.Java

```
package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import ModuloUsuario.ImgDoctor;
import javax.swing.table.DefaultTableModel;
public class BuscarAlumnoDerivacion extends javax.swing.JInternalFrame {
    public BuscarAlumnoDerivacion() {
        initComponents();
        ImgDoctor img = new ImgDoctor();
        jplImagen.add(img);
        jplImagen.repaint();
        cargarTabla();
    }
    private void cargarTabla(){
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_alumno", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        dtAlumno = alumno.GetTabla(colName, "alumno", "SELECT * FROM `alumno` ORDER BY `apellido`");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtAlumno, columnNames);
        jtAlumno.setModel(datos);
    }
    /*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")
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar= jtAlumno.getSelectedRow();
        if(filamodificar>=0){
            String apellido= jtAlumno.getValueAt(filamodificar, 0).toString();
            String nombre= jtAlumno.getValueAt(filamodificar, 1).toString();
            String dni= jtAlumno.getValueAt(filamodificar, 2).toString();
            DerivacionP.jlAlumno.setText(apellido);
            DerivacionP.jlAlumno1.setText(nombre);
            DerivacionP.jlAlumno2.setText(dni);
        }
        this.dispose();
    }
    private void jtAlumnoMouseClicked(java.awt.event.MouseEvent evt) {
        fila=jtAlumno.rowAtPoint(evt.getPoint());
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String texto= jtfBuscar.getText();
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_alumno", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        if (jrbDni.isSelected() == true) {
            dtAlumno = alumno.GetTabla(colName, "alumno", "SELECT * FROM `duenio` WHERE `dni_alumno`=" + texto + " ");
        }
        if (jrbApellido.isSelected() == true) {
            dtAlumno = alumno.GetTabla(colName, "alumno", "SELECT * FROM `doctor` WHERE `apellido` = \"" + texto + "\"");
        }
        DefaultTableModel datos = new DefaultTableModel(dtAlumno,columnNames);
        jtAlumno.setModel(datos); // se colocan los datos en la tabla
        //depurador de jtDuenio de item null
        int a = jtAlumno.getRowCount();
        int i=0;
        a--;
        while(a>=i){
            if(jtAlumno.getValueAt(i,1)==null){
                ((DefaultTableModel)jtAlumno.getModel()).removeRow(i);
                a--;
                i--;
            }
        }
    }
}
```

```
    }  
    i++;  
  }  
}  
Object[][] dtAlumno;  
MySQL alumno = new MySQL();  
int fila = -1;  
// Variables declaration - do not modify  
private javax.swing.ButtonGroup buttonGroup1;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JPanel jPanel2;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JButton jbBuscar;  
private javax.swing.JButton jbCancelar;  
private javax.swing.JButton jbElegir;  
private javax.swing.JPanel jPanelImagen;  
private javax.swing.JRadioButton jrbApellido;  
private javax.swing.JRadioButton jrbDni;  
public static javax.swing.JTable jtAlumno;  
private javax.swing.JTextField jtfBuscar;  
// End of variables declaration  
}
```

BuscarAlumnoDerivacionG.Java

```
package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import ModuloUsuario.ImgDoctor;
import javax.swing.table.DefaultTableModel;
public class BuscarAlumnoDerivacionG extends javax.swing.JInternalFrame {
    public BuscarAlumnoDerivacionG() {
        initComponents();
        ImgDoctor img = new ImgDoctor();
        jplImagen.add(img);
        jplImagen.repaint();
        cargarTabla();
    }
    private void cargarTabla(){
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_alumno", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        dtAlumno = alumno.GetTabla(colName, "alumno", "SELECT * FROM `alumno` ORDER BY `apellido`");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtAlumno, columnNames);
        jtAlumno.setModel(datos);
    }
    /* 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")
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar= jtAlumno.getSelectedRow();
        if(filamodificar>=0){
            String apellido= jtAlumno.getValueAt(filamodificar, 0).toString();
            String nombre= jtAlumno.getValueAt(filamodificar, 1).toString();
            String dni= jtAlumno.getValueAt(filamodificar, 2).toString();
            DerivacionP.jlAlumno.setText(apellido);
            DerivacionP.jlAlumno1.setText(nombre);
            DerivacionP.jlAlumno2.setText(dni);
        }
        this.dispose();
    }
    private void jtAlumnoMouseClicked(java.awt.event.MouseEvent evt) {
        fila=jtAlumno.rowAtPoint(evt.getPoint());
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String texto= jtfBuscar.getText();
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_alumno", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        if (jrbDni.isSelected() == true) {
            dtAlumno = alumno.GetTabla(colName, "alumno", "SELECT * FROM `duenio` WHERE `dni_alumno`=" + texto + " ");
        }
        if (jrbApellido.isSelected() == true) {
            dtAlumno = alumno.GetTabla(colName, "alumno", "SELECT * FROM `doctor` WHERE `apellido` = \"" + texto + "\"");
        }
        DefaultTableModel datos = new DefaultTableModel(dtAlumno, columnNames);
        jtAlumno.setModel(datos); // se colocan los datos en la tabla
        //depurador de jtDuenio de item null
        int a = jtAlumno.getRowCount();
        int i=0;
        a--;
        while(a>=i){
            if(jtAlumno.getValueAt(i,1)==null){
                ((DefaultTableModel)jtAlumno.getModel()).removeRow(i);
                a--;
                i--;
            }
        }
    }
}
```



```
    }  
    i++;  
  }  
}  
Object[][] dtAlumno;  
MySQL alumno = new MySQL();  
int fila = -1;  
// Variables declaration - do not modify  
private javax.swing.ButtonGroup buttonGroup1;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JPanel jPanel2;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JButton jbBuscar;  
private javax.swing.JButton jbCancelar;  
private javax.swing.JButton jbElegir;  
private javax.swing.JPanel jPanelImagen;  
private javax.swing.JRadioButton jrbApellido;  
private javax.swing.JRadioButton jrbDni;  
public static javax.swing.JTable jtAlumno;  
private javax.swing.JTextField jtfBuscar;  
// End of variables declaration  
}
```

BuscarAlumnoG.Java

```

package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import ModuloUsuario.ImgDoctor;
import javax.swing.table.DefaultTableModel;
public class BuscarAlumnoG extends javax.swing.JInternalFrame {
    public BuscarAlumnoG() {
        initComponents();
        ImgDoctor img = new ImgDoctor();
        jplImagen.add(img);
        jplImagen.repaint();
        cargarTabla();
    }
    private void cargarTabla(){
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_alumno", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        dtAlumno = alumno.GetTabla(colName, "alumno", "SELECT * FROM `alumno` ORDER BY `apellido`");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtAlumno, columnNames);
        jtAlumno.setModel(datos);
    }
    /*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")
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar= jtAlumno.getSelectedRow();
        if(filamodificar>=0){
            String apellido= jtAlumno.getValueAt(filamodificar, 0).toString();
            String nombre= jtAlumno.getValueAt(filamodificar, 1).toString();
            String dni= jtAlumno.getValueAt(filamodificar, 2).toString();
            ConsultaAnimalG.jlAlumno.setText(apellido);
            ConsultaAnimalG.jlAlumno1.setText(nombre);
            ConsultaAnimalG.jlAlumno2.setText(dni);
        }
        this.dispose();
    }
    private void jtAlumnoMouseClicked(java.awt.event.MouseEvent evt) {
        fila=jtAlumno.rowAtPoint(evt.getPoint());
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String texto= jtfBuscar.getText();
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_alumno", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        if (jrbDni.isSelected() == true) {
            dtAlumno = alumno.GetTabla(colName, "alumno", "SELECT * FROM `duenio` WHERE `dni_alumno`=" + texto + " ");
        }
        if (jrbApellido.isSelected() == true) {
            dtAlumno = alumno.GetTabla(colName, "alumno", "SELECT * FROM `doctor` WHERE `apellido` = \"" + texto + "\"");
        }
        DefaultTableModel datos = new DefaultTableModel(dtAlumno, columnNames);
        jtAlumno.setModel(datos); // se colocan los datos en la tabla
        //depurador de jtDuenio de item null
        int a = jtAlumno.getRowCount();
        int i=0;
        a--;
        while(a>=i){
            if(jtAlumno.getValueAt(i,1)==null){
                ((DefaultTableModel)jtAlumno.getModel()).removeRow(i);
                a--;
                i--;
            }
        }
    }
}

```

```
    }  
    i++;  
  }  
}  
Object[][] dtAlumno;  
MySQL alumno = new MySQL();  
int fila = -1;  
// Variables declaration - do not modify  
private javax.swing.ButtonGroup buttonGroup1;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JPanel jPanel2;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JButton jbBuscar;  
private javax.swing.JButton jbCancelar;  
private javax.swing.JButton jbElegir;  
private javax.swing.JPanel jPanelImagen;  
private javax.swing.JRadioButton jrbApellido;  
private javax.swing.JRadioButton jrbDni;  
public static javax.swing.JTable jtAlumno;  
private javax.swing.JTextField jtfBuscar;  
// End of variables declaration  
}
```

BuscarAlumnoRetorno.Java

```
package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import ModuloUsuario.ImgDoctor;
import javax.swing.table.DefaultTableModel;
public class BuscarAlumnoRetorno extends javax.swing.JFrame {
    public BuscarAlumnoRetorno() {
        initComponents();
        ImgDoctor img = new ImgDoctor();
        jPanel1.add(img);
        jPanel1.repaint();
        cargarTabla();
    }
    private void cargarTabla(){
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_alumno", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        dtAlumno = alumno.GetTabla(colName, "alumno", "SELECT * FROM `alumno` ORDER BY `apellido`");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtAlumno, columnNames);
        jtAlumno.setModel(datos);
    }
    /*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")
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar= jtAlumno.getSelectedRow();
        if(filamodificar>=0){
            String apellido= jtAlumno.getValueAt(filamodificar, 0).toString();
            String nombre= jtAlumno.getValueAt(filamodificar, 1).toString();
            String dni= jtAlumno.getValueAt(filamodificar, 2).toString();
            RetornoAnimalP.jlAlumno.setText(apellido);
            RetornoAnimalP.jlAlumno1.setText(nombre);
            RetornoAnimalP.jlAlumno2.setText(dni);
        }
        this.dispose();
    }
    private void jtAlumnoMouseClicked(java.awt.event.MouseEvent evt) {
        fila=jtAlumno.rowAtPoint(evt.getPoint());
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String texto= jtfBuscar.getText();
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_alumno", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        if (jrbDni.isSelected() == true) {
            dtAlumno = alumno.GetTabla(colName, "alumno", "SELECT * FROM `duenio` WHERE `dni_alumno`=" + texto + " ");
        }
        if (jrbApellido.isSelected() == true) {
            dtAlumno = alumno.GetTabla(colName, "alumno", "SELECT * FROM `doctor` WHERE `apellido` = \"" + texto + "\"");
        }
        DefaultTableModel datos = new DefaultTableModel(dtAlumno, columnNames);
        jtAlumno.setModel(datos); // se colocan los datos en la tabla
        //depurador de jtDuenio de item null
        int a = jtAlumno.getRowCount();
        int i=0;
        a--;
        while(a>=i){
            if(jtAlumno.getValueAt(i,1)==null){
                ((DefaultTableModel)jtAlumno.getModel()).removeRow(i);
                a--;
                i--;
            }
            i++;
        }
    }
}
```



```
}  
Object[][] dtAlumno;  
MySQL alumno = new MySQL();  
int fila = -1;  
// Variables declaration - do not modify  
private javax.swing.ButtonGroup buttonGroup1;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JPanel jPanel2;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JButton jbBuscar;  
private javax.swing.JButton jbCancelar;  
private javax.swing.JButton jbElegir;  
private javax.swing.JPanel jPanelImagen;  
private javax.swing.JRadioButton jrbApellido;  
private javax.swing.JRadioButton jrbDni;  
public static javax.swing.JTable jtAlumno;  
private javax.swing.JTextField jtfBuscar;  
// End of variables declaration  
}
```

BuscarAlumnoRetornoG.Java

```

package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import ModuloUsuario.ImgDoctor;
import javax.swing.table.DefaultTableModel;
public class BuscarAlumnoRetornoG extends javax.swing.JFrame {
    public BuscarAlumnoRetornoG() {
        initComponents();
        ImgDoctor img = new ImgDoctor();
        jplImagen.add(img);
        jplImagen.repaint();
        cargarTabla();
    }
    private void cargarTabla(){
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_alumno", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        dtAlumno = alumno.GetTabla(colName, "alumno", "SELECT * FROM `alumno` ORDER BY `apellido`");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtAlumno, columnNames);
        jtAlumno.setModel(datos);
    }
    /*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")
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar= jtAlumno.getSelectedRow();
        if(filamodificar>=0){
            String apellido= jtAlumno.getValueAt(filamodificar, 0).toString();
            String nombre= jtAlumno.getValueAt(filamodificar, 1).toString();
            String dni= jtAlumno.getValueAt(filamodificar, 2).toString();
            RetornoAnimalG.jlAlumno.setText(apellido);
            RetornoAnimalG.jlAlumno1.setText(nombre);
            RetornoAnimalG.jlAlumno2.setText(dni);
        }
        this.dispose();
    }
    private void jtAlumnoMouseClicked(java.awt.event.MouseEvent evt) {
        fila=jtAlumno.rowAtPoint(evt.getPoint());
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String texto= jtfBuscar.getText();
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_alumno", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        if (jrbDni.isSelected() == true) {
            dtAlumno = alumno.GetTabla(colName, "alumno", "SELECT * FROM `duenio` WHERE `dni_alumno`=" + texto + " ");
        }
        if (jrbApellido.isSelected() == true) {
            dtAlumno = alumno.GetTabla(colName, "alumno", "SELECT * FROM `doctor` WHERE `apellido` = \"" + texto + "\"");
        }
        DefaultTableModel datos = new DefaultTableModel(dtAlumno, columnNames);
        jtAlumno.setModel(datos); // se colocan los datos en la tabla
        //depurador de jtDuenio de item null
        int a = jtAlumno.getRowCount();
        int i=0;
        a--;
        while(a>=i){
            if(jtAlumno.getValueAt(i,1)==null){
                ((DefaultTableModel)jtAlumno.getModel()).removeRow(i);
                a--;
                i--;
            }
        }
    }
}

```



```
    }  
    i++;  
  }  
}  
Object[][] dtAlumno;  
MySQL alumno = new MySQL();  
int fila = -1;  
// Variables declaration - do not modify  
private javax.swing.ButtonGroup buttonGroup1;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JPanel jPanel2;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JButton jbBuscar;  
private javax.swing.JButton jbCancelar;  
private javax.swing.JButton jbElegir;  
private javax.swing.JPanel jPanelImagen;  
private javax.swing.JRadioButton jrbApellido;  
private javax.swing.JRadioButton jrbDni;  
public static javax.swing.JTable jtAlumno;  
private javax.swing.JTextField jtfBuscar;  
// End of variables declaration  
}
```

BuscarAnimalDerivacion.Java

```
package Consultas;
import ModuloAnimal.*;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import javax.swing.table.DefaultTableModel;
public class BuscarAnimalDerivacion extends javax.swing.JInternalFrame {
    public BuscarAnimalDerivacion() {
        initComponents();
        ImgAnimalp img = new ImgAnimalp();
        jplImagen.add(img);
        jplImagen.repaint();
        cargarTabla();
    }
    private void cargarTabla(){
        String[] columnNames = {"Apellido", "Nombre", "DNI", "Nombre", "Especie", "Raza", "Peso", "Fecha", "Sexo", "Color", "Vivo", "Codigo Animal"};
        String colName[] = {"apellido", "nombre", "dni_duenio", "animal_pequenos.nombre", "especie", "raza", "peso", "nacimiento", "sexo", "color", "vivo", "cod_animal"};
        //nombre de columnas, tabla, instruccion sql
        dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`. * FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
(`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio`) AND (`animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie`) AND
(`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza`)");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtAnimal, columnNames);
        jtAnimal.setModel(datos); // se colocan los datos en la tabla
        //depurador de jtDuenio de item null
        int a = jtAnimal.getRowCount();
        int i = 0;
        a--;
        while(a >= i){
            if(jtAnimal.getValueAt(i, 1) == null){
                ((DefaultTableModel)jtAnimal.getModel()).removeRow(i);
                a--;
                i--;
            }
            i++;
        }
    }
    /*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")
    private void jtAnimalMouseClicked(java.awt.event.MouseEvent evt) {
        fila = jtAnimal.rowAtPoint(evt.getPoint());
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String texto = jtfBuscar.getText();
        String[] columnNames = {"Apellido", "Nombre", "DNI", "Nombre", "Especie", "Raza", "Peso", "Fecha", "Sexo", "Color", "Vivo", "Codigo Animal"};
        String colName[] = {"apellido", "nombre", "dni_duenio", "animal_pequenos.nombre", "especie", "raza", "peso", "nacimiento", "sexo", "color", "vivo", "cod_animal"};
        //nombre de columnas, tabla, instruccion sql
        //nombre de columnas, tabla, instruccion sql
        if(jrbDni.isSelected() == true){
            dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`. * FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio` AND `animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie` AND
`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza` AND `duenio`.`dni_duenio` = " + texto + " ");
        }
        if(jrbNombre.isSelected() == true){
            dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`. * FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
(`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio`) AND (`animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie`) AND
(`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza`) AND (`animal_pequenos`.`nombre` = \" + texto + "\" )");
        }
    }
}
```

```
}
DefaultTableModel datos = new DefaultTableModel(dtAnimal,columnNames);
jtAnimal.setModel(datos); // se colocan los datos en la tabla
//depurador de jtDuenio de item null
int a = jtAnimal.getRowCount();
int i=0;
a--;
while(a>=i){
    if(jtAnimal.getValueAt(i,1)==null){
        ((DefaultTableModel)jtAnimal.getModel()).removeRow(i);
        a--;
        i--;
    }
    i++;
}
}
private void jbDetalleActionPerformed(java.awt.event.ActionEvent evt) {
    DetalleAnimal det=new DetalleAnimal();
    PantallaPrincipal.jdpFondo.add(det);
    det.show();
}
Object[][] dtAnimal;
Object[] registro;
MySQL animal = new MySQL();
int fila = -1;
// Variables declaration - do not modify
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JButton jbBuscar;
private javax.swing.JButton jbCancelar;
private javax.swing.JButton jbDetalle;
private javax.swing.JPanel jPanelImagen;
private javax.swing.JRadioButton jrbDni;
private javax.swing.JRadioButton jrbNombre;
public static javax.swing.JTable jtAnimal;
private javax.swing.JTextField jtFBuscar;
// End of variables declaration
}
```

BuscarAnimalDerivacionG.Java

```
package Consultas;
import ModuloAnimal.*;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import javax.swing.table.DefaultTableModel;
public class BuscarAnimalDerivacionG extends javax.swing.JFrame {
    public BuscarAnimalDerivacionG() {
        initComponents();
        ImgAnimalp img = new ImgAnimalp();
        jplImagen.add(img);
        jplImagen.repaint();
        cargarTabla();
    }
    private void cargarTabla(){
        String[] columnNames = {"Apellido", "Nombre", "DNI", "Nombre", "Especie", "Raza", "Peso", "Fecha", "Sexo", "Color", "Vivo", "Codigo Animal"};
        String colName[] = {"apellido", "nombre", "dni_duenio", "animal_pequenos.nombre", "especie", "raza", "peso", "nacimiento", "sexo", "color", "vivo", "cod_animal"};
        //nombre de columnas, tabla, instruccion sql
        dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`. *FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
(`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio`) AND (`animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie`) AND
(`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza`)");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtAnimal, columnNames);
        jtAnimal.setModel(datos); // se colocan los datos en la tabla
        //depurador de jtDuenio de item null
        int a = jtAnimal.getRowCount();
        int i = 0;
        a--;
        while(a >= i){
            if(jtAnimal.getValueAt(i, 1) == null){
                ((DefaultTableModel)jtAnimal.getModel()).removeRow(i);
                a--;
                i--;
            }
            i++;
        }
    }
    /*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")
    private void jtAnimalMouseClicked(java.awt.event.MouseEvent evt) {
        fila = jtAnimal.rowAtPoint(evt.getPoint());
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String texto = jtfBuscar.getText();
        String[] columnNames = {"Apellido", "Nombre", "DNI", "Nombre", "Especie", "Raza", "Peso", "Fecha", "Sexo", "Color", "Vivo", "Codigo Animal"};
        String colName[] = {"apellido", "nombre", "dni_duenio", "animal_pequenos.nombre", "especie", "raza", "peso", "nacimiento", "sexo", "color", "vivo", "cod_animal"};
        //nombre de columnas, tabla, instruccion sql
        //nombre de columnas, tabla, instruccion sql
        if(jrbDni.isSelected() == true){
            dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`. *FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio` AND `animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie` AND
`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza` AND `duenio`.`dni_duenio` = '" + texto + "'");
        }
        if(jrbNombre.isSelected() == true){
            dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`. *FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
(`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio`) AND (`animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie`) AND
(`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza`) AND (`animal_pequenos`.`nombre` = '" + texto + "')");
        }
    }
}
```

```
}
DefaultTableModel datos = new DefaultTableModel(dtAnimal,columnNames);
jtAnimal.setModel(datos); // se colocan los datos en la tabla
//depurador de jtDuenio de item null
int a = jtAnimal.getRowCount();
int i=0;
a--;
while(a>=i){
    if(jtAnimal.getValueAt(i,1)==null){
        ((DefaultTableModel)jtAnimal.getModel()).removeRow(i);
        a--;
        i--;
    }
    i++;
}
}
private void jbDetalleActionPerformed(java.awt.event.ActionEvent evt) {
    DetalleAnimal det=new DetalleAnimal();
    PantallaPrincipal.jdpFondo.add(det);
    det.show();
}
Object[][] dtAnimal;
Object[] registro;
MySQL animal = new MySQL();
int fila = -1;
// Variables declaration - do not modify
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JButton jbBuscar;
private javax.swing.JButton jbCancelar;
private javax.swing.JButton jbDetalle;
private javax.swing.JPanel jPanel3;
private javax.swing.JRadioButton jrbDni;
private javax.swing.JRadioButton jrbNombre;
public static javax.swing.JTable jtAnimal;
private javax.swing.JTextField jtfBuscar;
// End of variables declaration
}
```

BuscarCasoClinico.Java

```
package Consultas;
import Clases.MySQL;
import static Clases.PantallaPrincipal.jdpFondo;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.JOptionPane;
public class BuscarCasoClinico extends javax.swing.JInternalFrame {
    public BuscarCasoClinico() {
        initComponents();
        auxenviar.setVisible(false);
        auxenviar1.setVisible(false);
    }
    /* 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")
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String caso_clinico = tfCasoClinico.getText();
        SimpleDateFormat formatodeltexto = new SimpleDateFormat("yyyy");
        String fecha=tfCasoClinico1.getText();
        JOptionPane.showMessageDialog(rootPane, fecha);
        Date aux = null;
    try {
        aux = formatodeltexto.parse(fecha);
    } catch (ParseException ex) {
        ex.printStackTrace();
    }
    String colName[] = {"caso_clinico"};
    String fecha1=fecha+"0101";
    String fecha2=fecha+"1231";
    boolean a = consulta.existe2("consulta", "caso_clinico", "fecha", caso_clinico, fecha1, fecha2);
    if (a==false){
        JOptionPane.showMessageDialog(rootPane, "El Caso Clinico no existe");
    }
    else{
        this.dispose();
        auxenviar.setText(tfCasoClinico.getText());
        auxenviar1.setText(tfCasoClinico1.getText());
        RetornoAnimalP retorno = new RetornoAnimalP();
        jdpFondo.add(retorno);
        retorno.show();
    }
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    MySQL consulta = new MySQL();
    Object[][] dtconsulta;
    int fila = -1;
    // Variables declaration - do not modify
    public static javax.swing.JLabel auxenviar;
    public static javax.swing.JLabel auxenviar1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JButton jbBuscar;
    private javax.swing.JButton jbCancelar;
    private javax.swing.JTextField tfCasoClinico;
    private javax.swing.JTextField tfCasoClinico1;
    // End of variables declaration
}
```


BuscarCasoClinicoG.Java

```
package Consultas;
import Clases.MySQL;
import static Clases.PantallaPrincipal.jdpFondo;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.JOptionPane;
import javax.swing.JInternalFrame {
    public BuscarCasoClinicoG() {
        initComponents();
        auxenviarg.setVisible(false);
        auxenviarg2.setVisible(false); }
    /*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")
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String caso_clinico = tfCasoClinico.getText();
        SimpleDateFormat formatodeltexto = new SimpleDateFormat("yyyy");
        String fecha=tfCasoClinico1.getText();
        Date aux = null;
    try {
        aux = formatodeltexto.parse(fecha);
    } catch (ParseException ex) {
        ex.printStackTrace();
    }
    String colName[] = {"caso_clinico"};
    boolean a = consulta.existe("consultag", "caso_clinico", caso_clinico);
    if (a==false){
        JOptionPane.showMessageDialog(rootPane, "El Caso Clinico no existe");
    }else{
        String fecha1="2014-1-1";
        String fecha2="2014-12-31";
        dtconsulta = consulta.GetTabla(colName,"consultag", "SELECT * FROM consultag WHERE fecha BETWEEN "+fecha1+" AND "+fecha2+"");
        if (dtconsulta[0] == null){
            JOptionPane.showMessageDialog(rootPane, "El Caso Clinico no existe");
        }else{
            this.dispose();
            auxenviarg.setText(tfCasoClinico.getText());
            RetornoAnimalG retorno = new RetornoAnimalG();
            jdpFondo.add(retorno);
            retorno.show(); }
        }
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose(); }
    MySQL consulta = new MySQL();
    Object[][] dtconsulta;
    int fila = -1;
    // Variables declaration - do not modify
    public static javax.swing.JLabel auxenviarg;
    public static javax.swing.JLabel auxenviarg2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JButton jbBuscar;
    private javax.swing.JButton jbCancelar;
    private javax.swing.JTextField tfCasoClinico;
    private javax.swing.JTextField tfCasoClinico1;
    // End of variables declaration
}
```

BuscarCasoClinicoMostrarRetornoAG.Java

```
package Consultas;
import Clases.MySQL;
import static Clases.PantallaPrincipal.jdpFondo;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.JOptionPane;
public class BuscarCasoClinicoMostrarRetornoAG extends javax.swing.JFrame {
    public BuscarCasoClinicoMostrarRetornoAG() {
        initComponents();
        auxenviar.setVisible(false);
        auxenviar1.setVisible(false);
    }
    /*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")
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String caso_clinico = tfCasoClinico.getText();
        SimpleDateFormat formatodeltexto = new SimpleDateFormat("yyyy");
        String fecha=tfCasoClinico1.getText();
        JOptionPane.showMessageDialog(rootPane, fecha);
        Date aux = null;
    try {
        aux = formatodeltexto.parse(fecha);
    } catch (ParseException ex) {
        ex.printStackTrace();
    }
    String colName[] = {"caso_clinico"};
    String fecha1=fecha+"0101";
    String fecha2=fecha+"1231";
    JOptionPane.showMessageDialog(rootPane, "El Caso Clinico no existe");
    boolean a = consulta.existe2("consultag", "caso_clinico","fecha", caso_clinico,fecha1,fecha2);
    if (a==false){
        JOptionPane.showMessageDialog(rootPane, "El Caso Clinico no existe");
    }else{
        this.dispose();
        auxenviar.setText(tfCasoClinico.getText());
        auxenviar1.setText(tfCasoClinico1.getText());
        ListadoMostrarRetornoAG retorno = new ListadoMostrarRetornoAG();
        jdpFondo.add(retorno);
        retorno.show();
    }
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    MySQL consulta = new MySQL();
    Object[][] dtconsulta;
    int fila = -1;
    // Variables declaration - do not modify
    public static javax.swing.JLabel auxenviar;
    public static javax.swing.JLabel auxenviar1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JButton jbBuscar;
    private javax.swing.JButton jbCancelar;
    private javax.swing.JTextField tfCasoClinico;
    private javax.swing.JTextField tfCasoClinico1;
    // End of variables declaration
}
```

BuscarCasoClinicoMostrarRetornoAP.Java

```
package Consultas;
import Clases.MySQL;
import static Clases.PantallaPrincipal.jdpFondo;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.JOptionPane;
public class BuscarCasoClinicoMostrarRetornoAP extends javax.swing.JFrame {
    public BuscarCasoClinicoMostrarRetornoAP() {
        initComponents();
        auxenviar.setVisible(false);
        auxenviar1.setVisible(false);
    }
    /*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")
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String caso_clinico = tfCasoClinico.getText();
        SimpleDateFormat formatodeltexto = new SimpleDateFormat("yyyy");
        String fecha=tfCasoClinico1.getText();
        JOptionPane.showMessageDialog(rootPane, fecha);
        Date aux = null;
    try {
        aux = formatodeltexto.parse(fecha);
    } catch (ParseException ex) {
        ex.printStackTrace();
    }
        String colName[] = {"caso_clinico"};
        String fecha1=fecha+"0101";
        String fecha2=fecha+"1231";
        boolean a = consulta.existe2("consulta", "caso_clinico", "fecha", caso_clinico, fecha1, fecha2);
        if (a==false){
            JOptionPane.showMessageDialog(rootPane, "El Caso Clinico no existe");
        }else{
            this.dispose();
            auxenviar.setText(tfCasoClinico.getText());
            auxenviar1.setText(tfCasoClinico1.getText());
            ListadoMostrarRetornoAP retorno = new ListadoMostrarRetornoAP();
            jdpFondo.add(retorno);
            retorno.show();
        }
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    MySQL consulta = new MySQL();
    Object[][] dtconsulta;
    int fila = -1;
    // Variables declaration - do not modify
    public static javax.swing.JLabel auxenviar;
    public static javax.swing.JLabel auxenviar1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JButton jbBuscar;
    private javax.swing.JButton jbCancelar;
    private javax.swing.JTextField tfCasoClinico;
    private javax.swing.JTextField tfCasoClinico1;
    // End of variables declaration
}
```

BuscarDocente.Java

```
package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import ModuloUsuario.ImgDoctor;
import javax.swing.table.DefaultTableModel;
public class BuscarDocente extends javax.swing.JFrame {
    public BuscarDocente() {
        initComponents();
        ImgDoctor img = new ImgDoctor();
        jplImagen.add(img);
        jplImagen.repaint();
        cargarTabla();
    }
    private void cargarTabla(){
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_doctor", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        dtDoctor = doctor.GetTabla(colName, "doctor", "SELECT * FROM `doctor` ORDER BY `apellido`");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtDoctor, columnNames);
        jtDoctor.setModel(datos);
    }
    /*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")
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar= jtDoctor.getSelectedRow();
        if(filamodificar>=0){
            String apellido= jtDoctor.getValueAt(filamodificar, 0).toString();
            String nombre= jtDoctor.getValueAt(filamodificar, 1).toString();
            String dni= jtDoctor.getValueAt(filamodificar, 2).toString();
            ConsultaAnimalP.jlDocente.setText(apellido);
            ConsultaAnimalP.jlDocente1.setText(nombre);
            ConsultaAnimalP.jlDocente2.setText(dni);
        }
        this.dispose();
    }
    private void jtDoctorMouseClicked(java.awt.event.MouseEvent evt) {
        fila=jtDoctor.rowAtPoint(evt.getPoint());
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String texto= jtfBuscar.getText();
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_doctor", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        if(jrbDni.isSelected()==true){
            dtDoctor = doctor.GetTabla(colName, "doctor", "SELECT * FROM `doctor` WHERE `doctor`.`dni_doctor`='"+texto+"'");
        }
        if(jrbApellido.isSelected()==true){
            dtDoctor = doctor.GetTabla(colName, "doctor", "SELECT * FROM `doctor` WHERE `apellido` = '\"'+ texto + '\"");
        }
        DefaultTableModel datos = new DefaultTableModel(dtDoctor, columnNames);
        jtDoctor.setModel(datos); // se colocan los datos en la tabla
        //depurador de jtDuenio de item null
        int a = jtDoctor.getRowCount();
        int i=0;
        a--;
        while(a>=i){
            if(jtDoctor.getValueAt(i,1)==null){
                ((DefaultTableModel)jtDoctor.getModel()).removeRow(i);
                a--;
                i--;
            }
        }
    }
}
```

```
    }  
    i++;  
  }  
}  
Object[][] dtDoctor;  
MySQL doctor = new MySQL();  
int fila = -1;  
// Variables declaration - do not modify  
private javax.swing.ButtonGroup buttonGroup1;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JPanel jPanel2;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JButton jbBuscar;  
private javax.swing.JButton jbCancelar;  
private javax.swing.JButton jbElegir;  
private javax.swing.JPanel jPanelImagen;  
private javax.swing.JRadioButton jrbApellido;  
private javax.swing.JRadioButton jrbDni;  
public static javax.swing.JTable jtDoctor;  
private javax.swing.JTextField jtfBuscar;  
// End of variables declaration  
}
```

BuscarDocenteDerivacion.Java

```
package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import ModuloUsuario.ImgDoctor;
import javax.swing.table.DefaultTableModel;
public class BuscarDocenteDerivacion extends javax.swing.JInternalFrame {
    public BuscarDocenteDerivacion() {
        initComponents();
        ImgDoctor img = new ImgDoctor();
        jplImagen.add(img);
        jplImagen.repaint();
        cargarTabla();
    }
    private void cargarTabla(){
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_doctor", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        dtDoctor = doctor.GetTabla(colName, "doctor", "SELECT * FROM `doctor` ORDER BY `apellido`");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtDoctor, columnNames);
        jtDoctor.setModel(datos);
    }
    /*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")
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar= jtDoctor.getSelectedRow();
        if(filamodificar>=0){
            String apellido= jtDoctor.getValueAt(filamodificar, 0).toString();
            String nombre= jtDoctor.getValueAt(filamodificar, 1).toString();
            String dni= jtDoctor.getValueAt(filamodificar, 2).toString();
            DerivacionP.jlDocente.setText(apellido);
            DerivacionP.jlDocente1.setText(nombre);
            DerivacionP.jlDocente2.setText(dni);
        }
        this.dispose();
    }
    private void jtDoctorMouseClicked(java.awt.event.MouseEvent evt) {
        fila=jtDoctor.rowAtPoint(evt.getPoint());
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String texto= jtfBuscar.getText();
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_doctor", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        if(jrbDni.isSelected()==true){
            dtDoctor = doctor.GetTabla(colName, "doctor", "SELECT * FROM `doctor` WHERE `doctor`.`dni_doctor`='"+texto+"'");
        }
        if(jrbApellido.isSelected()==true){
            dtDoctor = doctor.GetTabla(colName, "doctor", "SELECT * FROM `doctor` WHERE `apellido` = '\" + texto + '\"");
        }
        DefaultTableModel datos = new DefaultTableModel(dtDoctor, columnNames);
        jtDoctor.setModel(datos); // se colocan los datos en la tabla
        //depurador de jtDuenio de item null
        int a = jtDoctor.getRowCount();
        int i=0;
        a--;
        while(a>=i){
            if(jtDoctor.getValueAt(i,1)==null){
                ((DefaultTableModel)jtDoctor.getModel()).removeRow(i);
                a--;
                i--;
            }
        }
    }
}
```

```
    }  
    i++;  
  }  
}  
Object[][] dtDoctor;  
MySQL doctor = new MySQL();  
int fila = -1;  
// Variables declaration - do not modify  
private javax.swing.ButtonGroup buttonGroup1;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JPanel jPanel2;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JButton jbBuscar;  
private javax.swing.JButton jbCancelar;  
private javax.swing.JButton jbElegir;  
private javax.swing.JPanel jPanelImagen;  
private javax.swing.JRadioButton jrbApellido;  
private javax.swing.JRadioButton jrbDni;  
public static javax.swing.JTable jtDoctor;  
private javax.swing.JTextField jtfBuscar;  
// End of variables declaration  
}
```


BuscarDocenteDerivacionG.Java

```

package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import ModuloUsuario.ImgDoctor;
import javax.swing.table.DefaultTableModel;
public class BuscarDocenteDerivacionG extends javax.swing.JInternalFrame {
    public BuscarDocenteDerivacionG() {
        initComponents();
        ImgDoctor img = new ImgDoctor();
        jplImagen.add(img);
        jplImagen.repaint();
        cargarTabla();
    }
    private void cargarTabla(){
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_doctor", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        dtDoctor = doctor.GetTabla(colName, "doctor", "SELECT * FROM `doctor` ORDER BY `apellido`");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtDoctor, columnNames);
        jtDoctor.setModel(datos);
    }
    /*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")
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar= jtDoctor.getSelectedRow();
        if(filamodificar>=0){
            String apellido= jtDoctor.getValueAt(filamodificar, 0).toString();
            String nombre= jtDoctor.getValueAt(filamodificar, 1).toString();
            String dni= jtDoctor.getValueAt(filamodificar, 2).toString();
            DerivacionG.jlDocente.setText(apellido);
            DerivacionG.jlDocente1.setText(nombre);
            DerivacionG.jlDocente2.setText(dni);
        }
        this.dispose();
    }
    private void jtDoctorMouseClicked(java.awt.event.MouseEvent evt) {
        fila=jtDoctor.rowAtPoint(evt.getPoint());
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String texto= jtfbBuscar.getText();
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_doctor", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        if(jrbDni.isSelected()){
            dtDoctor = doctor.GetTabla(colName, "doctor", "SELECT * FROM `doctor` WHERE `doctor`.`dni_doctor`='"+texto+"'");
        }
        if(jrbApellido.isSelected()){
            dtDoctor = doctor.GetTabla(colName, "doctor", "SELECT * FROM `doctor` WHERE `apellido` = '\"'+ texto + '\"");
        }
        DefaultTableModel datos = new DefaultTableModel(dtDoctor, columnNames);
        jtDoctor.setModel(datos); // se colocan los datos en la tabla
        //depurador de jtDuenio de item null
        int a = jtDoctor.getRowCount();
        int i=0;
        a--;
        while(a>=i){
            if(jtDoctor.getValueAt(i,1)==null){

```

```
((DefaultTableModel)jtDoctor.getModel()).removeRow(i);
a--;
i--;
}
i++;
}
}
Object[][] dtDoctor;
MySQL doctor = new MySQL();
int fila = -1;
// Variables declaration - do not modify
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JButton jbBuscar;
private javax.swing.JButton jbCancelar;
private javax.swing.JButton jbElegir;
private javax.swing.JPanel jPanelImagen;
private javax.swing.JRadioButton jrbApellido;
private javax.swing.JRadioButton jrbDni;
public static javax.swing.JTable jtDoctor;
private javax.swing.JTextField jtfBuscar;
// End of variables declaration
}
```

BuscarDocenteG.Java

```
package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import ModuloUsuario.ImgDoctor;
import javax.swing.table.DefaultTableModel;
public class BuscarDocenteG extends javax.swing.JInternalFrame {
    public BuscarDocenteG() {
        initComponents();
        ImgDoctor img = new ImgDoctor();
        jplImagen.add(img);
        jplImagen.repaint();
        cargarTabla();
    }
    private void cargarTabla(){
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_doctor", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        dtDoctor = doctor.GetTabla(colName, "doctor", "SELECT * FROM `doctor` ORDER BY `apellido`");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtDoctor, columnNames);
        jtDoctor.setModel(datos);
    }
    /*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")
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar= jtDoctor.getSelectedRow();
        if(filamodificar>=0){
            String apellido= jtDoctor.getValueAt(filamodificar, 0).toString();
            String nombre= jtDoctor.getValueAt(filamodificar, 1).toString();
            String dni= jtDoctor.getValueAt(filamodificar, 2).toString();
            ConsultaAnimalG.jlDocente.setText(apellido);
            ConsultaAnimalG.jlDocente1.setText(nombre);
            ConsultaAnimalG.jlDocente2.setText(dni);
        }
        this.dispose();
    }
    private void jtDoctorMouseClicked(java.awt.event.MouseEvent evt) {
        fila=jtDoctor.rowAtPoint(evt.getPoint());
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String texto= jtfBuscar.getText();
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_doctor", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        if(jrbDni.isSelected()==true){
            dtDoctor = doctor.GetTabla(colName, "doctor", "SELECT * FROM `doctor` WHERE `doctor`.`dni_doctor`='"+texto+"'");
        }
        if(jrbApellido.isSelected()==true){
            dtDoctor = doctor.GetTabla(colName, "doctor", "SELECT * FROM `doctor` WHERE `apellido` = '"+texto+"'");
        }
        DefaultTableModel datos = new DefaultTableModel(dtDoctor, columnNames);
        jtDoctor.setModel(datos); // se colocan los datos en la tabla
        //depurador de jtDuenio de item null
        int a = jtDoctor.getRowCount();
        int i=0;
        a--;
        while(a>=i){
            if(jtDoctor.getValueAt(i,1)==null){
                ((DefaultTableModel)jtDoctor.getModel()).removeRow(i);
                a--;
                i--;
            }
        }
    }
}
```

```
    }  
    i++;  
  }  
}  
Object[][] dtDoctor;  
MySQL doctor = new MySQL();  
int fila = -1;  
// Variables declaration - do not modify  
private javax.swing.ButtonGroup buttonGroup1;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JPanel jPanel2;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JButton jbBuscar;  
private javax.swing.JButton jbCancelar;  
private javax.swing.JButton jbElegir;  
private javax.swing.JPanel jPanelImagen;  
private javax.swing.JRadioButton jrbApellido;  
private javax.swing.JRadioButton jrbDni;  
public static javax.swing.JTable jtDoctor;  
private javax.swing.JTextField jtfBuscar;  
// End of variables declaration  
}
```

BuscarDocenteRetorno.Java

```
package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import ModuloUsuario.ImgDoctor;
import javax.swing.table.DefaultTableModel;
public class BuscarDocenteRetorno extends javax.swing.JInternalFrame {
    public BuscarDocenteRetorno() {
        initComponents();
        ImgDoctor img = new ImgDoctor();
        jplImagen.add(img);
        jplImagen.repaint();
        cargarTabla();
    }
    private void cargarTabla(){
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_doctor", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        dtDoctor = doctor.GetTabla(colName, "doctor", "SELECT * FROM `doctor` ORDER BY `apellido`");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtDoctor, columnNames);
        jtDoctor.setModel(datos);
    }
    /*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")
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar= jtDoctor.getSelectedRow();
        if(filamodificar>=0){
            String apellido= jtDoctor.getValueAt(filamodificar, 0).toString();
            String nombre= jtDoctor.getValueAt(filamodificar, 1).toString();
            String dni= jtDoctor.getValueAt(filamodificar, 2).toString();
            RetornoAnimalP.jlDocente.setText(apellido);
            RetornoAnimalP.jlDocente1.setText(nombre);
            RetornoAnimalP.jlDocente2.setText(dni);
        }
        this.dispose();
    }
    private void jtDoctorMouseClicked(java.awt.event.MouseEvent evt) {
        fila=jtDoctor.rowAtPoint(evt.getPoint());
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String texto= jtfBuscar.getText();
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_doctor", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        if(jrbDni.isSelected()==true){
            dtDoctor = doctor.GetTabla(colName, "doctor", "SELECT * FROM `doctor` WHERE `doctor`.`dni_doctor`='"+texto+"'");
        }
        if(jrbApellido.isSelected()==true){
            dtDoctor = doctor.GetTabla(colName, "doctor", "SELECT * FROM `doctor` WHERE `apellido` = '"+ texto + "'");
        }
        DefaultTableModel datos = new DefaultTableModel(dtDoctor,columnNames);
        jtDoctor.setModel(datos); // se colocan los datos en la tabla
        //depurador de jtDuenio de item null
        int a = jtDoctor.getRowCount();
        int i=0;
        a--;
        while(a>=i){
            if(jtDoctor.getValueAt(i,1)==null){
                ((DefaultTableModel)jtDoctor.getModel()).removeRow(i);
                a--;
                i--;
            }
        }
    }
}
```

```
    }  
    i++;  
  }  
}  
Object[][] dtDoctor;  
MySQL doctor = new MySQL();  
int fila = -1;  
// Variables declaration - do not modify  
private javax.swing.ButtonGroup buttonGroup1;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JPanel jPanel2;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JButton jbBuscar;  
private javax.swing.JButton jbCancelar;  
private javax.swing.JButton jbElegir;  
private javax.swing.JPanel jPanelImagen;  
private javax.swing.JRadioButton jrbApellido;  
private javax.swing.JRadioButton jrbDni;  
public static javax.swing.JTable jtDoctor;  
private javax.swing.JTextField jtfBuscar;  
// End of variables declaration  
}
```

BuscarDocenteRetornoG.Java

```
package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import ModuloUsuario.ImgDoctor;
import javax.swing.table.DefaultTableModel;
public class BuscarDocenteRetornoG extends javax.swing.JInternalFrame {
    public BuscarDocenteRetornoG() {
        initComponents();
        ImgDoctor img = new ImgDoctor();
        jplImagen.add(img);
        jplImagen.repaint();
        cargarTabla();
    }
    private void cargarTabla(){
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_doctor", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        dtDoctor = doctor.GetTabla(colName, "doctor", "SELECT * FROM `doctor` ORDER BY `apellido`");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtDoctor, columnNames);
        jtDoctor.setModel(datos);
    }
    /*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")
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar= jtDoctor.getSelectedRow();
        if(filamodificar>=0){
            String apellido= jtDoctor.getValueAt(filamodificar, 0).toString();
            String nombre= jtDoctor.getValueAt(filamodificar, 1).toString();
            String dni= jtDoctor.getValueAt(filamodificar, 2).toString();
            RetornoAnimalG.jlDocente.setText(apellido);
            RetornoAnimalG.jlDocente1.setText(nombre);
            RetornoAnimalG.jlDocente2.setText(dni);
        }
        this.dispose();
    }
    private void jtDoctorMouseClicked(java.awt.event.MouseEvent evt) {
        fila=jtDoctor.rowAtPoint(evt.getPoint());
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String texto= jtfBuscar.getText();
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_doctor", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        if(jrbDni.isSelected()==true){
            dtDoctor = doctor.GetTabla(colName, "doctor", "SELECT * FROM `doctor` WHERE `doctor`.`dni_doctor`='"+texto+"'");
        }
        if(jrbApellido.isSelected()==true){
            dtDoctor = doctor.GetTabla(colName, "doctor", "SELECT * FROM `doctor` WHERE `apellido` = '\" + texto + '\"");
        }
        DefaultTableModel datos = new DefaultTableModel(dtDoctor,columnNames);
        jtDoctor.setModel(datos); // se colocan los datos en la tabla
        //depurador de jtDuenio de item null
        int a = jtDoctor.getRowCount();
        int i=0;
        a--;
        while(a>=i){
            if(jtDoctor.getValueAt(i,1)==null){
                ((DefaultTableModel)jtDoctor.getModel()).removeRow(i);
                a--;
                i--;
            }
        }
    }
}
```



```
    }  
    i++;  
  }  
}  
Object[][] dtDoctor;  
MySQL doctor = new MySQL();  
int fila = -1;  
// Variables declaration - do not modify  
private javax.swing.ButtonGroup buttonGroup1;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JPanel jPanel2;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JButton jbBuscar;  
private javax.swing.JButton jbCancelar;  
private javax.swing.JButton jbElegir;  
private javax.swing.JPanel jPanelImagen;  
private javax.swing.JRadioButton jrbApellido;  
private javax.swing.JRadioButton jrbDni;  
public static javax.swing.JTable jtDoctor;  
private javax.swing.JTextField jtfBuscar;  
// End of variables declaration  
}
```

BuscarDueñoDerivacion.Java

```
package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import ModuloDueño.ImgDuenio;
import java.awt.Dimension;
import javax.swing.table.DefaultTableModel;
public class BuscarDueñoDerivacion extends javax.swing.JInternalFrame {
    public BuscarDueñoDerivacion() {
        initComponents();
        ImgDuenio img = new ImgDuenio();
        jplmagen.add(img);
        jplmagen.repaint();
        cargarTabla();
    }
    private void cargarTabla(){
        String[] columnNames = {"Apellido", "Nombre", "DNI", "Nombre", "Especie", "Raza", "Peso", "Fecha", "Sexo", "Color", "Vivo", "Codigo Animal"};
        String colName[] =
{"apellido", "nombre", "dni_duenio", "animal_pequenos.nombre", "especie", "raza", "peso", "nacimiento", "sexo", "color", "vivo", "cod_animal"};
//nombre de columnas, tabla, instruccion sql
        dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`.* FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
(`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio`) AND (`animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie`) AND
(`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza`)");
// se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtAnimal, columnNames);
        jtAnimal.setModel(datos); // se colocan los datos en la tabla
//depurador de jtDuenio de item null
        int a = jtAnimal.getRowCount();
        int i = 0;
        a--;
        while(a >= i){
            if(jtAnimal.getValueAt(i, 1) == null){
                ((DefaultTableModel)jtAnimal.getModel()).removeRow(i);
                a--;
                i--;
            }
            i++;
        }
    }
    /*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")
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar = jtAnimal.getSelectedRow();
        if(filamodificar >= 0){
            String apellido = jtAnimal.getValueAt(filamodificar, 0).toString();
            String dni = jtAnimal.getValueAt(filamodificar, 2).toString();
            String especie = jtAnimal.getValueAt(filamodificar, 4).toString();
            String raza = jtAnimal.getValueAt(filamodificar, 5).toString();
            String cod_animal = jtAnimal.getValueAt(filamodificar, 11).toString();
            DerivacionP.jlApellido.setText(apellido);
            DerivacionP.jlDni.setText(dni);
            DerivacionP.jlCodAnimal.setText(cod_animal);
            DerivacionP.jlDomicilio.setText(animal.GetData("domicilio", "SELECT domicilio FROM duenio WHERE dni_duenio = \"" + dni + "\""));
            DerivacionP.jlTelefono.setText(animal.GetData("telefono_fijo", "SELECT telefono_fijo FROM duenio WHERE dni_duenio = \"" + dni + "\""));
            DerivacionP.jlNombre.setText(animal.GetData("nombre", "SELECT nombre FROM animal_pequenos WHERE cod_animal = \"" + cod_animal + "\""));
            DerivacionP.jlFechaNacimiento.setText(animal.GetData("nacimiento", "SELECT nacimiento FROM animal_pequenos WHERE cod_animal =
\"" + cod_animal + "\""));
            DerivacionP.jlPeso.setText(animal.GetData("peso", "SELECT peso FROM animal_pequenos WHERE cod_animal = \"" + cod_animal + "\""));
            DerivacionP.jlSexo.setText(animal.GetData("sexo", "SELECT sexo FROM animal_pequenos WHERE cod_animal = \"" + cod_animal + "\""));
            DerivacionP.jlColor.setText(animal.GetData("color", "SELECT color FROM animal_pequenos WHERE cod_animal = \"" + cod_animal + "\""));
            DerivacionP.jlEspecie.setText(especie);
            DerivacionP.jlRaza.setText(raza);
            DerivacionP.jpGeneral.setPreferredSize(new Dimension(813, 259));
            this.dispose();
        }
    }
}
```

```

    }
}
private void jtAnimalMouseClicked(java.awt.event.MouseEvent evt) {
    fila=jtAnimal.rowAtPoint(evt.getPoint());
}
private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}
private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
    String texto= jtfBuscar.getText();
    String[] columnNames = {"Apellido", "Nombre", "DNI", "Nombre", "Especie", "Raza", "Peso", "Fecha", "Sexo", "Color", "Vivo", "Codigo Animal"};
    String colName[] =
{"apellido", "nombre", "dni_duenio", "animal_pequenos.nombre", "especie", "raza", "peso", "nacimiento", "sexo", "color", "vivo", "cod_animal"};
//nombre de columnas, tabla, instruccion sql
//nombre de columnas, tabla, instruccion sql
if(jrbDni.isSelected()==true){
    dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`. *FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio` AND `animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie` AND
`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza` AND `duenio`.`dni_duenio` = "+texto+" ");
}
if(jrbNombre.isSelected()==true){
    dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`. *FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
(`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio`) AND (`animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie`) AND
(`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza`) AND (`animal_pequenos`.`nombre` = \" + texto + \" \");
}
DefaultTableModel datos = new DefaultTableModel(dtAnimal, columnNames);
jtAnimal.setModel(datos); // se colocan los datos en la tabla
//depurador de jtDuenio de item null
int a = jtAnimal.getRowCount();
int i=0;
a--;
while(a>=i){
    if(jtAnimal.getValueAt(i,1)==null){
        ((DefaultTableModel)jtAnimal.getModel()).removeRow(i);
        a--;
        i--;
    }
    i++;
}
}
private void jbDetalleActionPerformed(java.awt.event.ActionEvent evt) {
    DetalleP det=new DetalleP();
    PantallaPrincipal.jdpFondo.add(det);
    det.show();
}
Object[][] dtAnimal;
MySQL animal = new MySQL();
int fila = -1;
// Variables declaration - do not modify
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JButton jbBuscar;
private javax.swing.JButton jbCancelar;
private javax.swing.JButton jbDetalle;
private javax.swing.JButton jbElegir;
private javax.swing.JPanel jPanelImagen;
private javax.swing.JRadioButton jrbDni;
private javax.swing.JRadioButton jrbNombre;
public static javax.swing.JTable jtAnimal;
private javax.swing.JTextField jtfBuscar;
// End of variables declaration
}

```

BuscarDueñoDerivacionG.Java

```

package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import ModuloDueño.ImgDuenio;
import java.awt.Dimension;
import javax.swing.table.DefaultTableModel;
public class BuscarDueñoDerivacionG extends javax.swing.JInternalFrame {
    public BuscarDueñoDerivacionG() {
        initComponents();
        ImgDuenio img = new ImgDuenio();
        jplmagen.add(img);
        jplmagen.repaint();
        cargarTabla();
    }
    private void cargarTabla(){
        String[] columnNames = {"Apellido", "Nombre", "DNI", "Nombre", "Especie", "Raza", "Peso", "Fecha", "Sexo", "Color", "Vivo", "Codigo Animal"};
        String colName[] =
{"apellido", "nombre", "dni_duenio", "animal_pequenos.nombre", "especie", "raza", "peso", "nacimiento", "sexo", "color", "vivo", "cod_animal"};
//nombre de columnas, tabla, instruccion sql
        dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`. * FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
(`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio`) AND (`animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie`) AND
(`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza`)");
// se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtAnimal, columnNames);
        jtAnimal.setModel(datos); // se colocan los datos en la tabla
//depurador de jtDuenio de item null
        int a = jtAnimal.getRowCount();
        int i = 0;
        a--;
        while(a >= i){
            if(jtAnimal.getValueAt(i, 1) == null){
                ((DefaultTableModel)jtAnimal.getModel()).removeRow(i);
                a--;
                i--;
            }
            i++;
        }
    }
    /*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")
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar = jtAnimal.getSelectedRow();
        if(filamodificar >= 0){
            String apellido = jtAnimal.getValueAt(filamodificar, 0).toString();
            String dni = jtAnimal.getValueAt(filamodificar, 2).toString();
            String especie = jtAnimal.getValueAt(filamodificar, 4).toString();
            String raza = jtAnimal.getValueAt(filamodificar, 5).toString();
            String cod_animal = jtAnimal.getValueAt(filamodificar, 11).toString();
            DerivacionG.jlCodAnimal.setText(cod_animal);
            DerivacionG.jlApellido.setText(apellido);
            DerivacionG.jlDni.setText(dni);
            DerivacionG.jlDomicilio.setText(animal.GetData("domicilio", "SELECT domicilio FROM duenio WHERE dni_duenio = \"" + dni + "\""));
            DerivacionG.jlTelefono.setText(animal.GetData("telefono_fijo", "SELECT telefono_fijo FROM duenio WHERE dni_duenio = \"" + dni + "\""));
            DerivacionG.jlNombre.setText(animal.GetData("nombre", "SELECT nombre FROM animal_pequenos WHERE cod_animal =
\"" + cod_animal + "\""));
            DerivacionG.jlFechaNacimiento.setText(animal.GetData("nacimiento", "SELECT nacimiento FROM animal_pequenos WHERE cod_animal =
\"" + cod_animal + "\""));
            DerivacionG.jlPeso.setText(animal.GetData("peso", "SELECT peso FROM animal_pequenos WHERE cod_animal = \"" + cod_animal + "\""));
            DerivacionG.jlSexo.setText(animal.GetData("sexo", "SELECT sexo FROM animal_pequenos WHERE cod_animal = \"" + cod_animal + "\""));
            DerivacionG.jlColor.setText(animal.GetData("color", "SELECT color FROM animal_pequenos WHERE cod_animal = \"" + cod_animal + "\""));
            DerivacionG.jlEspecie.setText(especie);
            DerivacionG.jlRaza.setText(raza);
            DerivacionG.jpGeneral.setPreferredSize(new Dimension(813, 259));
        }
    }
}

```

```

        this.dispose();
    }
}
private void jtAnimalMouseClicked(java.awt.event.MouseEvent evt) {
    fila=jtAnimal.rowAtPoint(evt.getPoint());
}
private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}
private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
    String texto= jtfBuscar.getText();
    String[] columnNames = {"Apellido", "Nombre", "DNI", "Nombre", "Especie", "Raza", "Peso", "Fecha", "Sexo", "Color", "Vivo", "Codigo Animal"};
    String colName[] =
{"apellido", "nombre", "dni_duenio", "animal_pequenos.nombre", "especie", "raza", "peso", "nacimiento", "sexo", "color", "vivo", "cod_animal"};
//nombre de columnas, tabla, instruccion sql
//nombre de columnas, tabla, instruccion sql
if(jrbDni.isSelected()==true){
    dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`. *FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio` AND `animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie` AND
`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza` AND `duenio`.`dni_duenio` = "+texto+" ");
}
if(jrbNombre.isSelected()==true){
    dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`. *FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
(`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio`) AND (`animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie`) AND
(`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza`) AND (`animal_pequenos`.`nombre` = \" + texto + "\") ");
}
DefaultTableModel datos = new DefaultTableModel(dtAnimal, columnNames);
jtAnimal.setModel(datos); // se colocan los datos en la tabla
//depurador de jtDuenio de item null
int a = jtAnimal.getRowCount();
int i=0;
a--;
while(a>=i){
    if(jtAnimal.getValueAt(i,1)==null){
        ((DefaultTableModel)jtAnimal.getModel()).removeRow(i);
        a--;
        i--;
    }
    i++;
}
}
private void jbDetalleActionPerformed(java.awt.event.ActionEvent evt) {
    DetalleP det=new DetalleP();
    PantallaPrincipal.jdpFondo.add(det);
    det.show();
}
Object[][] dtAnimal;
MySQL animal = new MySQL();
int fila = -1;
// Variables declaration - do not modify
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JButton jbBuscar;
private javax.swing.JButton jbCancelar;
private javax.swing.JButton jbDetalle;
private javax.swing.JButton jbElegir;
private javax.swing.JPanel jPanelImagen;
private javax.swing.JRadioButton jrbDni;
private javax.swing.JRadioButton jrbNombre;
public static javax.swing.JTable jtAnimal;
private javax.swing.JTextField jtfBuscar;
// End of variables declaration
}

```

BuscarDueñoG.Java

```

package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import ModuloDueño.ImgDuenio;
import java.awt.Dimension;
import javax.swing.table.DefaultTableModel;
public class BuscarDueñoG extends javax.swing.JFrame {
    public BuscarDueñoG() {
        initComponents();
        ImgDuenio img = new ImgDuenio();
        jplmagen.add(img);
        jplmagen.repaint();
        cargarTabla();
    }
    private void cargarTabla(){
        String[] columnNames = {"Apellido", "Nombre", "DNI", "Nombre", "Especie", "Raza", "Peso", "Fecha", "Sexo", "Color", "Vivo", "Codigo Animal"};
        String colName[] =
{"apellido", "nombre", "dni_duenio", "animal_pequenos.nombre", "especie", "raza", "peso", "nacimiento", "sexo", "color", "vivo", "cod_animal"};
//nombre de columnas, tabla, instruccion sql
        dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`.* FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
(`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio`) AND (`animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie`) AND
(`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza`)");
// se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtAnimal, columnNames);
        jtAnimal.setModel(datos); // se colocan los datos en la tabla
//depurador de jtDuenio de item null
        int a = jtAnimal.getRowCount();
        int i = 0;
        a--;
        while(a >= i){
            if(jtAnimal.getValueAt(i, 1) == null){
                ((DefaultTableModel)jtAnimal.getModel()).removeRow(i);
                a--;
                i--;
            }
            i++;
        }
    }
    /*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")
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar = jtAnimal.getSelectedRow();
        if(filamodificar >= 0){
            String apellido = jtAnimal.getValueAt(filamodificar, 0).toString();
            String dni = jtAnimal.getValueAt(filamodificar, 2).toString();
            String especie = jtAnimal.getValueAt(filamodificar, 4).toString();
            String raza = jtAnimal.getValueAt(filamodificar, 5).toString();
            String cod_animal = jtAnimal.getValueAt(filamodificar, 11).toString();
            ConsultaAnimalG.jlApellido.setText(apellido);
            ConsultaAnimalG.jlDni.setText(dni);
            ConsultaAnimalG.jlDomicilio.setText(animal.GetData("domicilio", "SELECT domicilio FROM duenio WHERE dni_duenio = \"" + dni + "\""));
            ConsultaAnimalG.jlTelefono.setText(animal.GetData("telefono_fijo", "SELECT telefono_fijo FROM duenio WHERE dni_duenio = \"" + dni + "\""));
            ConsultaAnimalG.jlNombre.setText(animal.GetData("nombre", "SELECT nombre FROM animal_pequenos WHERE cod_animal =
\"" + cod_animal + "\""));
            ConsultaAnimalG.jlFechaNacimiento.setText(animal.GetData("nacimiento", "SELECT nacimiento FROM animal_pequenos WHERE cod_animal =
\"" + cod_animal + "\""));
            ConsultaAnimalG.jlPeso.setText(animal.GetData("peso", "SELECT peso FROM animal_pequenos WHERE cod_animal = \"" + cod_animal + "\""));
            ConsultaAnimalG.jlSexo.setText(animal.GetData("sexo", "SELECT sexo FROM animal_pequenos WHERE cod_animal = \"" + cod_animal + "\""));
            ConsultaAnimalG.jlColor.setText(animal.GetData("color", "SELECT color FROM animal_pequenos WHERE cod_animal = \"" + cod_animal + "\""));
            ConsultaAnimalG.jlEspecie.setText(especie);
            ConsultaAnimalG.jlRaza.setText(raza);
            ConsultaAnimalG.jpGeneral.setPreferredSize(new Dimension(813, 259));
            this.dispose();
        }
    }
}

```



```

    }
}
private void jtAnimalMouseClicked(java.awt.event.MouseEvent evt) {
    fila=jtAnimal.rowAtPoint(evt.getPoint());
}
private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}
private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
    String texto= jtfBuscar.getText();
    String[] columNames = {"Apellido", "Nombre", "DNI", "Nombre", "Especie", "Raza", "Peso", "Fecha", "Sexo", "Color", "Vivo", "Codigo Animal"};
    String colName[] =
{"apellido", "nombre", "dni_duenio", "animal_pequenos.nombre", "especie", "raza", "peso", "nacimiento", "sexo", "color", "vivo", "cod_animal"};
//nombre de columnas, tabla, instruccion sql
//nombre de columnas, tabla, instruccion sql
if(jrbDni.isSelected()==true){
    dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`.*FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio` AND `animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie` AND
`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza` AND `duenio`.`dni_duenio` = "+texto+" ");
}
if(jrbNombre.isSelected()==true){

    dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`.*FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
(`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio`) AND (`animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie`) AND
(`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza`) AND (`animal_pequenos`.`nombre` = \" + texto + "\") ");
}
    DefaultTableModel datos = new DefaultTableModel(dtAnimal, columNames);
    jtAnimal.setModel(datos); // se colocan los datos en la tabla
//depurador de jtDuenio de item null
int a = jtAnimal.getRowCount();
int i=0;
a--;
while(a>=i){
    if(jtAnimal.getValueAt(i,1)==null){
        ((DefaultTableModel)jtAnimal.getModel()).removeRow(i);
        a--;
        i--;
    }
    i++;
}
}
private void jbDetalleActionPerformed(java.awt.event.ActionEvent evt) {
    DetalleP det=new DetalleP();
    PantallaPrincipal.jdpFondo.add(det);
    det.show();
}
Object[][] dtAnimal;
MySQL animal = new MySQL();
int fila = -1;
// Variables declaration - do not modify
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JButton jbBuscar;
private javax.swing.JButton jbCancelar;
private javax.swing.JButton jbDetalle;
private javax.swing.JButton jbElegir;
private javax.swing.JPanel jPanelImagen;
private javax.swing.JRadioButton jrbDni;
private javax.swing.JRadioButton jrbNombre;
public static javax.swing.JTable jtAnimal;
private javax.swing.JTextField jtfBuscar;
// End of variables declaration
}

```


BuscarDueñoP.Java

```
package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import ModuloDueño.ImgDuenio;
import java.awt.Dimension;
import javax.swing.table.DefaultTableModel;
public class BuscarDueñoP extends javax.swing.JInternalFrame {
    public BuscarDueñoP() {
        initComponents();
        ImgDuenio img = new ImgDuenio();
        jplmagen.add(img);
        jplmagen.repaint();
        cargarTabla();
    }
    private void cargarTabla(){
        String[] columnNames = {"Apellido", "Nombre", "DNI", "Nombre", "Especie", "Raza", "Peso", "Fecha", "Sexo", "Color", "Vivo", "Codigo Animal"};
        String colName[] =
{"apellido", "nombre", "dni_duenio", "animal_pequenos.nombre", "especie", "raza", "peso", "nacimiento", "sexo", "color", "vivo", "cod_animal"};
//nombre de columnas, tabla, instruccion sql
        dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`.* FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
(`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio`) AND (`animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie`) AND
(`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza`)");
// se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtAnimal, columnNames);
        jtAnimal.setModel(datos); // se colocan los datos en la tabla
//depurador de jtDuenio de item null
        int a = jtAnimal.getRowCount();
        int i = 0;
        a--;
        while(a >= i){
            if(jtAnimal.getValueAt(i, 1) == null){
                ((DefaultTableModel)jtAnimal.getModel()).removeRow(i);
                a--;
                i--;
            }
            i++;
        }
    }
    /*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")
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar = jtAnimal.getSelectedRow();
        if(filamodificar >= 0){
            String apellido = jtAnimal.getValueAt(filamodificar, 0).toString();
            String dni = jtAnimal.getValueAt(filamodificar, 2).toString();
            String especie = jtAnimal.getValueAt(filamodificar, 4).toString();
            String raza = jtAnimal.getValueAt(filamodificar, 5).toString();
            String cod_animal = jtAnimal.getValueAt(filamodificar, 11).toString();
            ConsultaAnimalP.jlApellido.setText(apellido);
            ConsultaAnimalP.jlDni.setText(dni);
            ConsultaAnimalP.jlDomicilio.setText(animal.GetData("domicilio", "SELECT domicilio FROM duenio WHERE dni_duenio = \"\"+dni+\"\""));
            ConsultaAnimalP.jlTelefono.setText(animal.GetData("telefono_fijo", "SELECT telefono_fijo FROM duenio WHERE dni_duenio = \"\"+dni+\"\""));
            ConsultaAnimalP.jlNombre.setText(animal.GetData("nombre", "SELECT nombre FROM animal_pequenos WHERE cod_animal =
\"\"+cod_animal+\"\""));
            ConsultaAnimalP.jlFechaNacimiento.setText(animal.GetData("nacimiento", "SELECT nacimiento FROM animal_pequenos WHERE cod_animal =
\"\"+cod_animal+\"\""));
            ConsultaAnimalP.jlPeso.setText(animal.GetData("peso", "SELECT peso FROM animal_pequenos WHERE cod_animal = \"\"+cod_animal+\"\""));
            ConsultaAnimalP.jlSexo.setText(animal.GetData("sexo", "SELECT sexo FROM animal_pequenos WHERE cod_animal = \"\"+cod_animal+\"\""));
            ConsultaAnimalP.jlColor.setText(animal.GetData("color", "SELECT color FROM animal_pequenos WHERE cod_animal = \"\"+cod_animal+\"\""));
            ConsultaAnimalP.jlEspecie.setText(especie);
            ConsultaAnimalP.jlRaza.setText(raza);
            ConsultaAnimalP.jpGeneral.setPreferredSize(new Dimension(813, 259));
            this.dispose();
        }
    }
}
```

```

    }
}
private void jtAnimalMouseClicked(java.awt.event.MouseEvent evt) {
    fila=jtAnimal.rowAtPoint(evt.getPoint());
}
private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}
private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
    String texto= jtfBuscar.getText();
    String[] columnNames = {"Apellido", "Nombre", "DNI", "Nombre", "Especie", "Raza", "Peso", "Fecha", "Sexo", "Color", "Vivo", "Codigo Animal"};
    String colName[] =
{"apellido", "nombre", "dni_duenio", "animal_pequenos.nombre", "especie", "raza", "peso", "nacimiento", "sexo", "color", "vivo", "cod_animal"};
//nombre de columnas, tabla, instruccion sql
//nombre de columnas, tabla, instruccion sql
if(jrbDni.isSelected()==true){
    dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`.*FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio` AND `animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie` AND
`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza` AND `duenio`.`dni_duenio` = "+texto+" ");
}
if(jrbNombre.isSelected()==true){
    dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`.*FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
(`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio`) AND (`animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie`) AND
(`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza`) AND (`animal_pequenos`.`nombre` = \" + texto + "\") ");
}
DefaultTableModel datos = new DefaultTableModel(dtAnimal, columnNames);
jtAnimal.setModel(datos); // se colocan los datos en la tabla
//depurador de jtDuenio de item null
int a = jtAnimal.getRowCount();
int i=0;
a--;
while(a>=i){
    if(jtAnimal.getValueAt(i,1)==null){
        ((DefaultTableModel)jtAnimal.getModel()).removeRow(i);
        a--;
        i--;
    }
    i++;
}
}
private void jbDetalleActionPerformed(java.awt.event.ActionEvent evt) {
    DetalleP det=new DetalleP();
    PantallaPrincipal.jdpFondo.add(det);
    det.show();
}
Object[][] dtAnimal;
MySQL animal = new MySQL();
int fila = -1;
// Variables declaration - do not modify
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JButton jbBuscar;
private javax.swing.JButton jbCancelar;
private javax.swing.JButton jbDetalle;
private javax.swing.JButton jbElegir;
private javax.swing.JPanel jPanelImagen;
private javax.swing.JRadioButton jrbDni;
private javax.swing.JRadioButton jrbNombre;
public static javax.swing.JTable jtAnimal;
private javax.swing.JTextField jtfBuscar;
// End of variables declaration
}

```

BuscarDueñoRetorno.Java

```
package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import ModuloDueño.ImgDuenio;
import java.awt.Dimension;
import javax.swing.table.DefaultTableModel;
public class BuscarDueñoRetorno extends javax.swing.JInternalFrame {
    public BuscarDueñoRetorno() {
        initComponents();
        ImgDuenio img = new ImgDuenio();
        jplmagen.add(img);
        jplmagen.repaint();
        cargarTabla();
    }
    private void cargarTabla(){
        String[] columnNames = {"Apellido", "Nombre", "DNI", "Nombre", "Especie", "Raza", "Peso", "Fecha", "Sexo", "Color", "Vivo", "Codigo Animal"};
        String colName[] =
{"apellido", "nombre", "dni_duenio", "animal_pequenos.nombre", "especie", "raza", "peso", "nacimiento", "sexo", "color", "vivo", "cod_animal"};
//nombre de columnas, tabla, instruccion sql
        dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`. * FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
(`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio`) AND (`animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie`) AND
(`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza`)");
// se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtAnimal, columnNames);
        jtAnimal.setModel(datos); // se colocan los datos en la tabla
//depurador de jtDuenio de item null
        int a = jtAnimal.getRowCount();
        int i = 0;
        a--;
        while(a >= i){
            if(jtAnimal.getValueAt(i, 1) == null){
                ((DefaultTableModel)jtAnimal.getModel()).removeRow(i);
                a--;
                i--;
            }
            i++;
        }
    }
    /*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")
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar = jtAnimal.getSelectedRow();
        if(filamodificar >= 0){
            String apellido = jtAnimal.getValueAt(filamodificar, 0).toString();
            String dni = jtAnimal.getValueAt(filamodificar, 2).toString();
            String especie = jtAnimal.getValueAt(filamodificar, 4).toString();
            String raza = jtAnimal.getValueAt(filamodificar, 5).toString();
            String cod_animal = jtAnimal.getValueAt(filamodificar, 11).toString();
            RetornoAnimalP.jlApellido.setText(apellido);
            RetornoAnimalP.jlDni.setText(dni);
            RetornoAnimalP.jlDomicilio.setText(animal.GetData("domicilio", "SELECT domicilio FROM duenio WHERE dni_duenio = \"" + dni + "\""));
            RetornoAnimalP.jlTelefono.setText(animal.GetData("telefono_fijo", "SELECT telefono_fijo FROM duenio WHERE dni_duenio = \"" + dni + "\""));
            RetornoAnimalP.jlNombre.setText(animal.GetData("nombre", "SELECT nombre FROM animal_pequenos WHERE cod_animal =
\"" + cod_animal + "\""));
            RetornoAnimalP.jlFechaNacimiento.setText(animal.GetData("nacimiento", "SELECT nacimiento FROM animal_pequenos WHERE cod_animal =
\"" + cod_animal + "\""));
            RetornoAnimalP.jlPeso.setText(animal.GetData("peso", "SELECT peso FROM animal_pequenos WHERE cod_animal = \"" + cod_animal + "\""));
            RetornoAnimalP.jlSexo.setText(animal.GetData("sexo", "SELECT sexo FROM animal_pequenos WHERE cod_animal = \"" + cod_animal + "\""));
            RetornoAnimalP.jlColor.setText(animal.GetData("color", "SELECT color FROM animal_pequenos WHERE cod_animal = \"" + cod_animal + "\""));
            RetornoAnimalP.jlEspecie.setText(especie);
            RetornoAnimalP.jlRaza.setText(raza);
            RetornoAnimalP.jpGeneral.setPreferredSize(new Dimension(813, 259));
            this.dispose();
        }
    }
}
```

```

    }
}
private void jtAnimalMouseClicked(java.awt.event.MouseEvent evt) {
    fila=jtAnimal.rowAtPoint(evt.getPoint());
}
private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}
private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
    String texto= jtfBuscar.getText();
    String[] columnNames = {"Apellido", "Nombre", "DNI", "Nombre", "Especie", "Raza", "Peso", "Fecha", "Sexo", "Color", "Vivo", "Codigo Animal"};
    String colName[] =
{"apellido", "nombre", "dni_duenio", "animal_pequenos.nombre", "especie", "raza", "peso", "nacimiento", "sexo", "color", "vivo", "cod_animal"};
//nombre de columnas, tabla, instruccion sql
//nombre de columnas, tabla, instruccion sql
if(jrbDni.isSelected()==true){
    dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`.*FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio` AND `animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie` AND
`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza` AND `duenio`.`dni_duenio` = "+texto+" ");
}
if(jrbNombre.isSelected()==true){
    dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`.*FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
(`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio`) AND (`animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie`) AND
(`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza`) AND (`animal_pequenos`.`nombre` = \""+ texto + "\"");
}
DefaultTableModel datos = new DefaultTableModel(dtAnimal, columnNames);
jtAnimal.setModel(datos); // se colocan los datos en la tabla
//depurador de jtDuenio de item null
int a = jtAnimal.getRowCount();
int i=0;
a--;
while(a>=i){
    if(jtAnimal.getValueAt(i,1)==null){
        ((DefaultTableModel)jtAnimal.getModel()).removeRow(i);
        a--;
        i--;
    }
    i++;
}
}
private void jbDetalleActionPerformed(java.awt.event.ActionEvent evt) {
    DetalleP det=new DetalleP();
    PantallaPrincipal.jdpFondo.add(det);
    det.show();
}
Object[][] dtAnimal;
MySQL animal = new MySQL();
int fila = -1;
// Variables declaration - do not modify
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JButton jbBuscar;
private javax.swing.JButton jbCancelar;
private javax.swing.JButton jbDetalle;
private javax.swing.JButton jbElegir;
private javax.swing.JPanel jPanelImagen;
private javax.swing.JRadioButton jrbDni;
private javax.swing.JRadioButton jrbNombre;
public static javax.swing.JTable jtAnimal;
private javax.swing.JTextField jtfBuscar;
// End of variables declaration
}

```

```

package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import ModuloDueño.ImgDuenio;
import java.awt.Dimension;
import javax.swing.table.DefaultTableModel;
public class BuscarDueñoRetornoG extends javax.swing.JInternalFrame {
    public BuscarDueñoRetornoG() {
        initComponents();
        ImgDuenio img = new ImgDuenio();
        jplImagen.add(img);
        jplImagen.repaint();
        cargarTabla();
    }
    private void cargarTabla(){
        String[] columnNames = {"Apellido", "Nombre", "DNI", "Nombre", "Especie", "Raza", "Peso", "Fecha", "Sexo", "Color", "Vivo", "Codigo Animal"};
        String colName[] =
{"apellido", "nombre", "dni_duenio", "animal_pequenos.nombre", "especie", "raza", "peso", "nacimiento", "sexo", "color", "vivo", "cod_animal"};
//nombre de columnas, tabla, instruccion sql
        dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`. * FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
(`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio`) AND (`animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie`) AND
(`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza`)");
// se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtAnimal, columnNames);
        jtAnimal.setModel(datos); // se colocan los datos en la tabla
//depurador de jtDuenio de item null
        int a = jtAnimal.getRowCount();
        int i = 0;
        a--;
        while(a >= i){
            if(jtAnimal.getValueAt(i, 1) == null){
                ((DefaultTableModel)jtAnimal.getModel()).removeRow(i);
                a--;
                i--;
            }
            i++;
        }
    }
    /*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")
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar = jtAnimal.getSelectedRow();
        if(filamodificar >= 0){
            String apellido = jtAnimal.getValueAt(filamodificar, 0).toString();
            String dni = jtAnimal.getValueAt(filamodificar, 2).toString();
            String especie = jtAnimal.getValueAt(filamodificar, 4).toString();
            String raza = jtAnimal.getValueAt(filamodificar, 5).toString();
            String cod_animal = jtAnimal.getValueAt(filamodificar, 11).toString();
            RetornoAnimalG.jlApellido.setText(apellido);
            RetornoAnimalG.jlDni.setText(dni);
            RetornoAnimalG.jlDomicilio.setText(animal.GetData("domicilio", "SELECT domicilio FROM duenio WHERE dni_duenio = \"" + dni + "\""));
            RetornoAnimalG.jlTelefono.setText(animal.GetData("telefono_fijo", "SELECT telefono_fijo FROM duenio WHERE dni_duenio = \"" + dni + "\""));
            RetornoAnimalG.jlNombre.setText(animal.GetData("nombre", "SELECT nombre FROM animal_pequenos WHERE cod_animal =
\"" + cod_animal + "\""));
            RetornoAnimalG.jlFechaNacimiento.setText(animal.GetData("nacimiento", "SELECT nacimiento FROM animal_pequenos WHERE cod_animal =
\"" + cod_animal + "\""));
            RetornoAnimalG.jlPeso.setText(animal.GetData("peso", "SELECT peso FROM animal_pequenos WHERE cod_animal = \"" + cod_animal + "\""));
            RetornoAnimalG.jlSexo.setText(animal.GetData("sexo", "SELECT sexo FROM animal_pequenos WHERE cod_animal = \"" + cod_animal + "\""));
            RetornoAnimalG.jlColor.setText(animal.GetData("color", "SELECT color FROM animal_pequenos WHERE cod_animal = \"" + cod_animal + "\""));
            RetornoAnimalG.jlEspecie.setText(especie);
            RetornoAnimalG.jlRaza.setText(raza);
            RetornoAnimalG.jpGeneral.setPreferredSize(new Dimension(813, 259));
            this.dispose();
        }
    }
    private void jtAnimalMouseClicked(java.awt.event.MouseEvent evt) {
        fila = jtAnimal.rowAtPoint(evt.getPoint());
    }
}

```

```

private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}
private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
    String texto= jtfBuscar.getText();
    String[] columnNames = {"Apellido", "Nombre", "DNI", "Nombre", "Especie", "Raza", "Peso", "Fecha", "Sexo", "Color", "Vivo", "Codigo Animal"};
    String colName[] =
{"apellido", "nombre", "dni_duenio", "animal_pequenos.nombre", "especie", "raza", "peso", "nacimiento", "sexo", "color", "vivo", "cod_animal"};
//nombre de columnas, tabla, instruccion sql
//nombre de columnas, tabla, instruccion sql
if(jrbDni.isSelected()==true){
    dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`. *FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio` AND `animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie` AND
`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza` AND `duenio`.`dni_duenio` = "+texto+" ");
}
if(jrbNombre.isSelected()==true){
    dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`. *FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
(`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio`) AND (`animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie`) AND
(`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza`) AND (`animal_pequenos`.`nombre` = \" + texto + \" \") ");
}
DefaultTableModel datos = new DefaultTableModel(dtAnimal, columnNames);
jtAnimal.setModel(datos); // se colocan los datos en la tabla
//depurador de jtDuenio de item null
int a = jtAnimal.getRowCount();
int i=0;
a--;
while(a>=i){
    if(jtAnimal.getValueAt(i,1)==null){
        ((DefaultTableModel)jtAnimal.getModel()).removeRow(i);
        a--;
        i--;
    }
    i++;
}
}
private void jbDetalleActionPerformed(java.awt.event.ActionEvent evt) {
    DetalleP det=new DetalleP();
    PantallaPrincipal.jdpFondo.add(det);
    det.show();
}
Object[][] dtAnimal;
MySQL animal = new MySQL();
int fila = -1;
// Variables declaration - do not modify
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JButton jbBuscar;
private javax.swing.JButton jbCancelar;
private javax.swing.JButton jbDetalle;
private javax.swing.JButton jbElegir;
private javax.swing.JPanel jPanelmagen;
private javax.swing.JRadioButton jrbDni;
private javax.swing.JRadioButton jrbNombre;
public static javax.swing.JTable jtAnimal;
private javax.swing.JTextField jtfBuscar;
// End of variables declaration
}

```


BuscarMostrarCasoClinico.Java

```
package Consultas;
import Clases.MySQL;
import static Clases.PantallaPrincipal.jdpFondo;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.JOptionPane;
public class BuscarMostrarCasoClinico extends javax.swing.JFrame {
    public BuscarMostrarCasoClinico() {
        initComponents();
        auxenviar.setVisible(false);
        auxenviar1.setVisible(false);
    }
    /* 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")
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String caso_clinico = tfCasoClinico.getText();
        String fecha=tfCasoClinico1.getText();
        String colName[] = {"caso_clinico"};
        String fecha1=fecha+"0101";
        String fecha2=fecha+"1231";
        boolean a = consulta.existe2("consulta", "caso_clinico","fecha", caso_clinico,fecha1,fecha2);
        if (a==false){
            JOptionPane.showMessageDialog(rootPane, "El Caso Clinico no existe");
        }else{
            this.dispose();
            auxenviar.setText(tfCasoClinico.getText());
            auxenviar1.setText(tfCasoClinico1.getText());
            MostrarConsultaAnimalP retorno = new MostrarConsultaAnimalP();
            jdpFondo.add(retorno);
            retorno.show();
        }
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    MySQL consulta = new MySQL();
    Object[][] dtconsulta;
    int fila = -1;
    // Variables declaration - do not modify
    public static javax.swing.JLabel auxenviar;
    public static javax.swing.JLabel auxenviar1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JButton jbBuscar;
    private javax.swing.JButton jbCancelar;
    private javax.swing.JTextField tfCasoClinico;
    private javax.swing.JTextField tfCasoClinico1;
    // End of variables declaration
}
```


BuscarMostrarCascoClinicoG.Java

```
package Consultas;
import Clases.MySQL;
import static Clases.PantallaPrincipal.jdpFondo;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.JOptionPane;
public class BuscarMostrarCasoClinicoG extends javax.swing.JFrame {
    public BuscarMostrarCasoClinicoG() {
        initComponents();
        auxenviar.setVisible(false);
        auxenviar1.setVisible(false);
    }
    /* 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")
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String caso_clinico = tfCasoClinico.getText();
        String fecha=tfCasoClinico1.getText();
        String colName[] = {"caso_clinico"};
        String fecha1=fecha+"0101";
        String fecha2=fecha+"1231";
        boolean a = consulta.existe2("consultag", "caso_clinico", "fecha", caso_clinico, fecha1, fecha2);
        if (a==false){
            JOptionPane.showMessageDialog(rootPane, "El Caso Clinico no existe"+a+"2");
        }else{
            JOptionPane.showMessageDialog(rootPane, "El Caso Clinico"+a);
            this.dispose();
            auxenviar.setText(tfCasoClinico.getText());
            auxenviar1.setText(tfCasoClinico1.getText());
            MostrarConsultaAnimalG retorno = new MostrarConsultaAnimalG();
            jdpFondo.add(retorno);
            retorno.show();
        }
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    MySQL consulta = new MySQL();
    Object[][] dtconsulta;
    int fila = -1;
    // Variables declaration - do not modify
    public static javax.swing.JLabel auxenviar;
    public static javax.swing.JLabel auxenviar1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JButton jbBuscar;
    private javax.swing.JButton jbCancelar;
    private javax.swing.JTextField tfCasoClinico;
    private javax.swing.JTextField tfCasoClinico1;
    // End of variables declaration
}
```

ConsultaAnimalG.Java

```

package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import java.sql.Date;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
public class ConsultaAnimalG extends javax.swing.JInternalFrame {
    public ConsultaAnimalG() {
        initComponents();
        Calendar Cal= Calendar.getInstance();
        int Max= Integer.parseInt(animal.GetData("caso_clinico", "SELECT MAX(caso_clinico) caso_clinico FROM consultag"));
        int num= Max +1;
        jxCasoClinico.setText(""+num+"/"+Cal.get(Cal.YEAR));
        String fec= Cal.get(Cal.DATE)+"/"+(Cal.get(Cal.MONTH)+1)+"/"+Cal.get(Cal.YEAR);
        String hor=Cal.get(Cal.HOUR_OF_DAY)+":"+Cal.get(Cal.MINUTE)+":"+Cal.get(Cal.SECOND);
        jlFecha.setText(fec);
        jlHora.setText(hor);
    }
    /* 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")
    private void jbGuardarActionPerformed(java.awt.event.ActionEvent evt) {
        String caso_clinico = jxCasoClinico.getText();
        String cod_animal="1";
        String dni_doctor=jlDocente2.getText();
        String dni_alumno=jlAlumno2.getText();
        boolean aux = animal.existe("consultag", "caso_clinico", caso_clinico);
        if (aux==true){
            JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
        }else{
            String motivo = jtaMotivo.getText();
            String anamnesis = jtaAnamnesis.getText();
            String temperatura = jtftemperatura.getText();
            String fc = jtffc.getText();
            String fr = jtffr.getText();
            String cc = (String)jcbCC.getSelectedItemAt();
            String ll_cap = (String)jcbLLCap.getSelectedItemAt();
            String pulso_ritmo = (String)jcbPulsoRitmo.getSelectedItemAt();
            String pulso_intensidad = (String)jcbPulsoIntensidad.getSelectedItemAt();
            String hidratacion2="Deshidratado";
            String hidratacion=(String)jcbHidratacion.getSelectedItemAt();
            if (hidratacion2==hidratacion){
                hidratacion = jtfdeshidratado.getText();
            }else{
                hidratacion=(String)jcbHidratacion.getSelectedItemAt();
            }
            String ma_ocular = (String)jcbMaOcular.getSelectedItemAt();
            String ma_bucal = (String)jcbMaBucal.getSelectedItemAt();
            String ma_nasal = (String)jcbMaNasal.getSelectedItemAt();
            String ma_genital = (String)jcbMaGenital.getSelectedItemAt();
            String submandibular = (String)jcbSubmandibular1.getSelectedItemAt();
            String submandibular2 = "No";
            if(submandibular==submandibular2){
                submandibular =
                (String)jcbSubmandibular2.getSelectedItemAt()+"", "+jcbSubmandibular3.getSelectedItemAt()+"", "+jcbSubmandibular4.getSelectedItemAt();
            }else{
                submandibular = "Si";
            }
        }
    }

```

```

String axilar = (String)jcbAxilar1.getSelectedItemAt();
String axilar2 = "No";
if(axilar==axilar2){
    axilar = (String)jcbAxilar2.getSelectedItemAt()+" "+jcbAxilar3.getSelectedItemAt()+" "+jcbAxilar4.getSelectedItemAt();
}else{
    axilar = "Si";
}
String inguinales = (String)jcbInguinales1.getSelectedItemAt();
String inguinales2 = "No";
if(inguinales==inguinales2){
    inguinales = (String)jcbInguinales2.getSelectedItemAt()+" "+jcbInguinales3.getSelectedItemAt()+" "+jcbInguinales4.getSelectedItemAt();
}else{
    inguinales = "Si";
}
String popliteos = (String)jcbPopliteos1.getSelectedItemAt();
String popliteos2 = "No";
if(popliteos==popliteos2){
    popliteos = (String)jcbPopliteos2.getSelectedItemAt()+" "+jcbPopliteos3.getSelectedItemAt()+" "+jcbPopliteos4.getSelectedItemAt();
}else{
    popliteos = "Si";
}
String hallazgos = jtaHallazgos.getText();
String diagnostico_presuntivo = jtaDiagnosticoPresuntivo.getText();
String diagnostico_diferencial = jtaDiagnosticoDiferencial.getText();
String diagnostico_pronostico = jtaDiagnosticoPronostico.getText();
String tratamiento = jtaTratamiento.getText();
String indicaciones = jtaIndicaciones.getText();
String lesion_ubicacion="dds";
java.util.Date date = new java.util.Date();
java.text.SimpleDateFormat sdf=new java.text.SimpleDateFormat("yyyy-MM-dd");
String fecha = sdf.format(date);
String datos[] =
{caso_clinico,fecha,cod_animal,dni_doctor,dni_alumno,motivo,anamnesis,temperatura,fc,fr,cc,ll_cap,pulso_ritmo,pulso_intensidad,hidratacion,ma_
ocular,ma_bucal,ma_nasal,ma_genital,submandibular,axilar,inguinales,popliteos,hallazgos,diagnostico_presuntivo,diagnostico_diferencial,diagnostico_
o_pronostico,tratamiento,indicaciones};
    animal.InsertarRegistro(datos, "insert into consultag values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");
}
}
private void jBuscarPropietarioActionPerformed(java.awt.event.ActionEvent evt) {
    BuscarDueñoG buscar = new BuscarDueñoG();
    PantallaPrincipal.jdpFondo.add(buscar);
    buscarToFront();
    buscar.setVisible(true);
}
private void jcbDatosDueñoActionPerformed(java.awt.event.ActionEvent evt) {
    String dni= jIDni.getText();
    String dato = jcbDatosDueño.getSelectedItemAt().toString();
    if(dato=="e-mail"){
        jIDato.setText(animal.GetData("email","SELECT email FROM duenio WHERE dni_duenio = \"'+dni+'\""));
    }
    if(dato=="facebook"){
        jIDato.setText(animal.GetData("facebook","SELECT facebook FROM duenio WHERE dni_duenio = \"'+dni+'\""));
    }
    if(dato=="celular"){
        jIDato.setText(animal.GetData("telefono_celular","SELECT telefono_celular FROM duenio WHERE dni_duenio = \"'+dni+'\""));
    }
}
private void jcbPopliteos1ActionPerformed(java.awt.event.ActionEvent evt) {
    String popliteos = (String)jcbPopliteos1.getSelectedItemAt();
    String popliteos2 = "No";
    if(popliteos==popliteos2){
        jpPopliteos.setVisible(true);
    }else{
        jpPopliteos.setVisible(false);
    }
}
private void jcbInguinales1ActionPerformed(java.awt.event.ActionEvent evt) {
    String inguinales = (String)jcbInguinales1.getSelectedItemAt();
    String inguinales2 = "No";
    if(inguinales==inguinales2){
        jpInguinales.setVisible(true);
    }
}

```

```

    }else{
        jpInguinales.setVisible(false);
    }
}

private void jcbAxilar1ActionPerformed(java.awt.event.ActionEvent evt) {
    String axilar = (String)jcbAxilar1.getSelectedItem();
    String axilar2 = "No";
    if(axilar==axilar2){
        jpAxilar.setVisible(true);
    }else{
        jpAxilar.setVisible(false);
    }
}

private void jcbSubmandibular1ActionPerformed(java.awt.event.ActionEvent evt) {
    String submandibular = (String)jcbSubmandibular1.getSelectedItem();
    String submandibular2 = "No";
    if(submandibular==submandibular2){
        jpSubmandibular.setVisible(true);
    }else{
        jpSubmandibular.setVisible(false);
    }
}

private void jcbHidratacionActionPerformed(java.awt.event.ActionEvent evt) {
    String hidratado="Deshidratado";
    String hidratado2=(String)jcbHidratacion.getSelectedItem();
    if (hidratado2==hidratado){
        jpDeshidratado.setVisible(true);
    }else{
        jpDeshidratado.setVisible(false);
    }
}

private void jbSalirActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}

private void jBuscarDocenteActionPerformed(java.awt.event.ActionEvent evt) {
    BuscarDocenteG buscarDocenteG = new BuscarDocenteG();
    PantallaPrincipal.jdpFondo.add(buscarDocenteG);
    buscarDocenteG.toFront();
    buscarDocenteG.setVisible(true);
}

private void jBuscarAlumnoActionPerformed(java.awt.event.ActionEvent evt) {
    BuscarAlumnoG buscarAlumnoG = new BuscarAlumnoG();
    PantallaPrincipal.jdpFondo.add(buscarAlumnoG);
    buscarAlumnoG.toFront();
    buscarAlumnoG.setVisible(true);
}

MySQL animal = new MySQL();
// Variables declaration - do not modify
public javax.swing.JButton jBuscarAlumno;
public javax.swing.JButton jBuscarDocente;
public javax.swing.JButton jBuscarPropietario;
public javax.swing.JLabel jLabel1;
public javax.swing.JLabel jLabel10;
public javax.swing.JLabel jLabel11;
public javax.swing.JLabel jLabel116;
public javax.swing.JLabel jLabel117;
public javax.swing.JLabel jLabel118;
public javax.swing.JLabel jLabel119;
public javax.swing.JLabel jLabel12;
public javax.swing.JLabel jLabel120;
public javax.swing.JLabel jLabel121;
public javax.swing.JLabel jLabel122;
public javax.swing.JLabel jLabel123;
public javax.swing.JLabel jLabel124;
public javax.swing.JLabel jLabel125;
public javax.swing.JLabel jLabel13;
public javax.swing.JLabel jLabel14;
public javax.swing.JLabel jLabel143;
public javax.swing.JLabel jLabel15;
public static javax.swing.JLabel jLabel16;
public javax.swing.JLabel jLabel17;

```

```
public javax.swing.JLabel jLabel18;
public javax.swing.JLabel jLabel19;
public javax.swing.JLabel jLabel2;
public javax.swing.JLabel jLabel20;
public javax.swing.JLabel jLabel21;
public javax.swing.JLabel jLabel22;
public javax.swing.JLabel jLabel23;
public javax.swing.JLabel jLabel24;
public javax.swing.JLabel jLabel25;
public javax.swing.JLabel jLabel26;
public javax.swing.JLabel jLabel27;
public javax.swing.JLabel jLabel28;
public javax.swing.JLabel jLabel29;
public javax.swing.JLabel jLabel3;
public javax.swing.JLabel jLabel30;
public javax.swing.JLabel jLabel31;
public javax.swing.JLabel jLabel32;
public javax.swing.JLabel jLabel33;
public javax.swing.JLabel jLabel34;
public javax.swing.JLabel jLabel35;
public javax.swing.JLabel jLabel36;
public javax.swing.JLabel jLabel37;
public javax.swing.JLabel jLabel38;
public javax.swing.JLabel jLabel39;
public javax.swing.JLabel jLabel4;
public javax.swing.JLabel jLabel40;
public javax.swing.JLabel jLabel41;
public javax.swing.JLabel jLabel42;
public javax.swing.JLabel jLabel43;
public javax.swing.JLabel jLabel44;
public javax.swing.JLabel jLabel45;
public javax.swing.JLabel jLabel46;
public javax.swing.JLabel jLabel47;
public javax.swing.JLabel jLabel48;
public javax.swing.JLabel jLabel5;
public javax.swing.JLabel jLabel6;
public javax.swing.JLabel jLabel7;
public javax.swing.JLabel jLabel8;
public javax.swing.JLabel jLabel9;
public javax.swing.JScrollPane jScrollPane1;
public javax.swing.JScrollPane jScrollPane2;
public javax.swing.JScrollPane jScrollPane3;
public javax.swing.JScrollPane jScrollPane31;
public javax.swing.JScrollPane jScrollPane32;
public javax.swing.JScrollPane jScrollPane33;
public javax.swing.JScrollPane jScrollPane34;
public javax.swing.JScrollPane jScrollPane35;
public javax.swing.JScrollPane jScrollPane36;
public static javax.swing.JScrollPane jScrollPane37;
public static javax.swing.JTabbedPane jTabbedPane1;
public javax.swing.JButton jButtonGuardar;
public javax.swing.JButton jButtonSalir;
public javax.swing.JComboBox jcbAxilar1;
public javax.swing.JComboBox jcbAxilar2;
public javax.swing.JComboBox jcbAxilar3;
public javax.swing.JComboBox jcbAxilar4;
public javax.swing.JComboBox jcbCC;
public javax.swing.JComboBox jcbDatosDueño;
public javax.swing.JComboBox jcbHidratacion;
public javax.swing.JComboBox jcbInguinales1;
public javax.swing.JComboBox jcbInguinales2;
public javax.swing.JComboBox jcbInguinales3;
public javax.swing.JComboBox jcbInguinales4;
public javax.swing.JComboBox jcbLLCap;
public javax.swing.JComboBox jcbMaBucal;
public javax.swing.JComboBox jcbMaGenital;
public javax.swing.JComboBox jcbMaNasal;
public javax.swing.JComboBox jcbMaOcular;
public javax.swing.JComboBox jcbPopliteos1;
public javax.swing.JComboBox jcbPopliteos2;
public javax.swing.JComboBox jcbPopliteos3;
```

```
public javax.swing.JComboBox jcbPopliteos4;
public javax.swing.JComboBox jcbPulsoIntensidad;
public javax.swing.JComboBox jcbPulsoRitmo;
public javax.swing.JComboBox jcbSubmandibular1;
public javax.swing.JComboBox jcbSubmandibular2;
public javax.swing.JComboBox jcbSubmandibular3;
public javax.swing.JComboBox jcbSubmandibular4;
public static javax.swing.JLabel jlAlumno;
public static javax.swing.JLabel jlAlumno1;
public static javax.swing.JLabel jlAlumno2;
public static javax.swing.JLabel jlApellido;
public javax.swing.JLabel jlCasoClinico;
public static javax.swing.JLabel jlColor;
public javax.swing.JLabel jlDato;
public static javax.swing.JLabel jlDni;
public static javax.swing.JLabel jlDocente;
public static javax.swing.JLabel jlDocente1;
public static javax.swing.JLabel jlDocente2;
public static javax.swing.JLabel jlDomicilio;
public static javax.swing.JLabel jlEspecie;
public javax.swing.JLabel jlFecha;
public static javax.swing.JLabel jlFechaNacimiento;
public javax.swing.JLabel jlHora;
public static javax.swing.JLabel jlNombre;
public static javax.swing.JLabel jlPeso;
public static javax.swing.JLabel jlRaza;
public static javax.swing.JLabel jlSexo;
public static javax.swing.JLabel jlTelefono;
public javax.swing.JPanel jpAxilar;
public javax.swing.JPanel jpConsulta1;
public javax.swing.JPanel jpConsulta10;
public javax.swing.JPanel jpConsulta2;
public javax.swing.JPanel jpConsulta5;
public javax.swing.JPanel jpDeshidratado;
public static javax.swing.JPanel jpGeneral;
public javax.swing.JPanel jpInguinales;
public javax.swing.JPanel jpPopliteos;
public javax.swing.JPanel jpSubmandibular;
public javax.swing.JTextArea jtaAnamnesis;
public javax.swing.JTextArea jtaCodExámenes;
public javax.swing.JTextArea jtaDiagnosticoDiferencial;
public javax.swing.JTextArea jtaDiagnosticoPresuntivo;
public javax.swing.JTextArea jtaDiagnosticoPronostico;
public javax.swing.JTextArea jtaHallazgos;
public javax.swing.JTextArea jtaIndicaciones;
public javax.swing.JTextArea jtaMotivo;
public javax.swing.JTextArea jtaTratamiento;
public javax.swing.JTextField jtfDeshidratado;
public javax.swing.JTextField jtfFC;
public javax.swing.JTextField jtfFR;
public javax.swing.JTextField jtfTemperatura;
public javax.swing.JTabbedPane jtpPestañas;
// End of variables declaration
}
```


ConsultaAnimalP.Java

```

package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import java.beans.PropertyVetoException;
import java.sql.Date;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFrame;
import javax.swing.JInternalFrame;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
public class ConsultaAnimalP extends javax.swing.JInternalFrame {
    public ConsultaAnimalP() {
        initComponents();
        jpSecreciones.setVisible(false);
        jpLesion.setVisible(false);
        jpGenitalesSecreciones.setVisible(false);
        Calendar Cal= Calendar.getInstance();
        int Max= Integer.parseInt(animal.GetData("caso_clinico", "SELECT MAX(caso_clinico) caso_clinico FROM consulta"));
        int num= Max +1;
        jlcasoClinico.setText(""+num+"/"+Cal.get(Calendar.YEAR));
        String fec= Cal.get(Calendar.DATE)+"/"+(Cal.get(Calendar.MONTH)+1)+"/"+Cal.get(Calendar.YEAR);
        String hor=Cal.get(Calendar.HOUR_OF_DAY)+":"+Cal.get(Calendar.MINUTE)+":"+Cal.get(Calendar.SECOND);
        jlFecha.setText(fec);
        jlHora.setText(hor);
    }
    /* 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")
    private void jcbGenitalesSecrecionesActionPerformed(java.awt.event.ActionEvent evt) {
        String genitalesSecreciones = (String)jcbGenitalesSecreciones.getSelectedItem();
        String genitalesSecreciones2 = "Si";
        if(genitalesSecreciones==genitalesSecreciones2){
            jpGenitalesSecreciones.setVisible(true);
        }else{
            jpGenitalesSecreciones.setVisible(false);
        }
    }
    private void jtfLesionFormaActionPerformed(java.awt.event.ActionEvent evt) {
    }
    private void jcbHidratacionActionPerformed(java.awt.event.ActionEvent evt) {
        String hidratado="Deshidratado";
        String hidratado2=(String)jcbHidratacion.getSelectedItem();
        if (hidratado2==hidratado){
            jpDeshidratado.setVisible(true);
        }else{
            jpDeshidratado.setVisible(false);
        }
    }
    private void jcbSubmandibular1ActionPerformed(java.awt.event.ActionEvent evt) {
        String submandibular = (String)jcbSubmandibular1.getSelectedItem();
        String submandibular2 = "No";
        if(submandibular==submandibular2){
            jpSubmandibular.setVisible(true);
        }else{
            jpSubmandibular.setVisible(false);
        }
    }
    private void jcbAxilar1ActionPerformed(java.awt.event.ActionEvent evt) {
        String axilar = (String)jcbAxilar1.getSelectedItem();
    }
}

```



```
String axilar2 = "No";
if(axilar==axilar2){
    jpAxilar.setVisible(true);
}else{
    jpAxilar.setVisible(false);
}
}

private void jcbInguinales1ActionPerformed(java.awt.event.ActionEvent evt) {
    String inguinales = (String)jcbInguinales1.getSelectedItem();
    String inguinales2 = "No";
    if(inguinales==inguinales2){
        jpInguinales.setVisible(true);
    }else{
        jpInguinales.setVisible(false);
    }
}

private void jcbPopliteos1ActionPerformed(java.awt.event.ActionEvent evt) {
    String popliteos = (String)jcbPopliteos1.getSelectedItem();
    String popliteos2 = "No";
    if(popliteos==popliteos2){
        jpPopliteos.setVisible(true);
    }else{
        jpPopliteos.setVisible(false);
    }
}

private void jcbSecrecionesActionPerformed(java.awt.event.ActionEvent evt) {
    String secreciones = (String)jcbSecreciones.getSelectedItem();
    String secreciones2 = "Si";
    if(secreciones==secreciones2){
        jpSecreciones.setVisible(true);
    }else{
        jpSecreciones.setVisible(false);
    }
}

private void jcbLesionActionPerformed(java.awt.event.ActionEvent evt) {
    String lesion = (String)jcbLesion.getSelectedItem();
    String lesion2 = "Si";
    if(lesion==lesion2){
        jpLesion.setVisible(true);
    }else{
        jpLesion.setVisible(false);
    }
}

private void jcbClaudicacion1ActionPerformed(java.awt.event.ActionEvent evt) {
    String claudicacion = (String)jcbClaudicacion1.getSelectedItem();
    String claudicacion2 = "Si";
    if(claudicacion==claudicacion2){
        jpClaudicacion.setVisible(true);
    }else{
        jpClaudicacion.setVisible(false);
    }
}

private void jcbParalisisActionPerformed(java.awt.event.ActionEvent evt) {
    String paralisis = (String)jcbParalisis.getSelectedItem();
    String paralisis2 = "Si";
    if(paralisis==paralisis2){
        jpParalisis.setVisible(true);
    }else{
        jpParalisis.setVisible(false);
    }
}

private void jcbParesiaActionPerformed(java.awt.event.ActionEvent evt) {
    String paresia = (String)jcbParesia.getSelectedItem();
    String paresia2 = "Si";
    if(paresia ==paresia2){
        jpParesia.setVisible(true);
    }else{
        jpParesia.setVisible(false);
    }
}

private void jcbConvulsionActionPerformed(java.awt.event.ActionEvent evt) {
```

```

String convulsion = (String)jcbConvulsion.getSelectedItemAt();
String convulsion2 = "Si";
if(convulsion ==convulsion2){
    jpConvulsion.setVisible(true);
}else{
    jpConvulsion.setVisible(false);
}
}
private void jbGuardarActionPerformed(java.awt.event.ActionEvent evt) {
    int Max= Integer.parseInt(animal.GetData("caso_clinico", "SELECT MAX(caso_clinico) caso_clinico FROM consulta"));
    int num= Max +1;
    String caso_clinico = String.valueOf(num);
    String cod_animal="1";
    String dni_doctor=jIDocente2.getText();
    String dni_alumno=jIAlumno2.getText();
    String lesion_tipo="";
    String lesion_forma="";
    String lesion_simetria="";
    String olor="";
    String prurito="";
    String manto_piloso="";
    String ectoparasitos="";
    String oido="";
    boolean aux = animal.existe("consulta", "caso_clinico", caso_clinico);
    if (aux==true){
        JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
    }else{
        String motivo = jtaMotivo.getText();
        String anamnesis = jtaAnamnesis.getText();
        String temperatura = jtftemperatura.getText();
        String fc = jtffc.getText();
        String fr = jtffr.getText();
        String cc = (String)jcbCC.getSelectedItemAt();
        String ll_cap = (String)jcbLLCap.getSelectedItemAt();
        String pulso_ritmo = (String)jcbPulsoRitmo.getSelectedItemAt();
        String pulso_intensidad = (String)jcbPulsoIntensidad.getSelectedItemAt();
        String hidratacion2="Deshidratado";
        String hidratacion=(String)jcbHidratacion.getSelectedItemAt();
        if (hidratacion2==hidratacion){
            hidratacion = jtfdeshidratado.getText();
        }else{
            hidratacion=(String)jcbHidratacion.getSelectedItemAt();
        }
        String ma_ocular = (String)jcbMaOcular.getSelectedItemAt();
        String ma_bucal = (String)jcbMaBucal.getSelectedItemAt();
        String ma_nasal = (String)jcbMaNasal.getSelectedItemAt();
        String ma_genital = (String)jcbMaGenital.getSelectedItemAt();
        String submandibular = (String)jcbSubmandibular1.getSelectedItemAt();
        String submandibular2 = "No";
        if(submandibular==submandibular2){
            submandibular =
            (String)jcbSubmandibular2.getSelectedItemAt()+" "+jcbSubmandibular3.getSelectedItemAt()+" "+jcbSubmandibular4.getSelectedItemAt();
        }else{
            submandibular = "Si";
        }
        String axilar = (String)jcbAxilar1.getSelectedItemAt();
        String axilar2 = "No";
        if(axilar==axilar2){
            axilar = (String)jcbAxilar2.getSelectedItemAt()+" "+jcbAxilar3.getSelectedItemAt()+" "+jcbAxilar4.getSelectedItemAt();
        }else{
            axilar = "Si";
        }
        String inguinales = (String)jcbInguinales1.getSelectedItemAt();
        String inguinales2 = "No";
        if(inguinales==inguinales2){
            inguinales = (String)jcbInguinales2.getSelectedItemAt()+" "+jcbInguinales3.getSelectedItemAt()+" "+jcbInguinales4.getSelectedItemAt();
        }else{
            inguinales = "Si";
        }
        String popliteos = (String)jcbPopliteos1.getSelectedItemAt();
        String popliteos2 = "No";
    }
}

```

```

if(popliteos==popliteos2){
    popliteos = (String)jcbPopliteos2.getSelectedItemAt()+","+jcbPopliteos3.getSelectedItemAt()+","+jcbPopliteos4.getSelectedItemAt();
}else{
    popliteos = "Si";
}
String lesion = (String)jcbLesion.getSelectedItemAt();
String lesion2 = "Si";
if(lesion==lesion2){
    lesion_tipo = jtaLesionTipo.getText();
    lesion_forma = jtaLesionForma.getText();
    lesion_simetria = (String)jcbLesionSimetria.getSelectedItemAt();
    olor = jtaLesionOlor.getText();
    prurito = (String)jcbLesionPrurito.getSelectedItemAt();
    manto_piloso = jtaLesionManto.getText();
    ectoparasitos = (String)jcbLesionEctoparasitos.getSelectedItemAt();
    oido = jtaLesionOidos.getText();
}
String boca = jtaBoca.getText();
String esofago = jtaEsofago.getText();
String estomago = jtaEstomago.getText();
String intestino = jtaIntestinos.getText();
String higado = jtaHigado.getText();
String regurgitacion = (String)jcbRegurgitacion.getSelectedItemAt();
String vomito = (String)jcbVomito.getSelectedItemAt();
String diarrea = (String)jcbDiarrea.getSelectedItemAt();
String ruidos = (String)jcbRuidos.getSelectedItemAt();
String distension = (String)jcbDistension.getSelectedItemAt();
String vias_superiores = jtaViasSuperiores.getText();
String vias_inferiores = jtaViasInferiores.getText();
String ritmo_respiratorio = (String)jcbRitmoRespiratorio.getSelectedItemAt();
String tipo = (String)jcbTipo.getSelectedItemAt();
String tos_reflejo = (String)jcbTosReflejo.getSelectedItemAt();
String auscultacion = jtaAuscultacion.getText();
String riñones = jtaRiñones.getText();
String ureteres = jtaUreteres.getText();
String vejiga = jtaVejiga.getText();
String uretra = jtaUretra.getText();
String secreciones = (String)jcbSecreciones.getSelectedItemAt();
String secreciones2 = "Si";
if(secreciones==secreciones2){
    secreciones = jtaSecreciones.getText();
}else{
    secreciones = "";
}
String orina = jtaOrina.getText();
String corazon_auscultacion = jtaCorazonAuscultacion.getText();
String ritmo_corazon = (String)jcbRitmoCorazon.getSelectedItemAt();
String genitales_interno = jtaGenitalesInterno.getText();
String genitales_externo = jtaGenitalesExterno.getText();
String genitales_secreciones = (String)jcbGenitalesSecreciones.getSelectedItemAt();
String genitales_secreciones2 = "Si";
if(genitales_secreciones==genitales_secreciones2){
    genitales_secreciones=jtaGenitalesSecreciones.getText();
}
String tacto_rectal = jtaTactoRectal.getText();
String locomotor_lesion = jtaLocomotorLesion.getText();
String locomotor_ubicacionL = jtaLocomotorUbicacionL.getText();
String locomotor_deformacion = jtaLocomotorDesformacion.getText();
String locomotor_ubicacionD = jtaLocomotorUbicacionD.getText();
String claudicacion_tipo = (String)jcbClaudicacion1.getSelectedItemAt();
String claudicacion_tipo2 = "Si";
if(claudicacion_tipo==claudicacion_tipo2){
    claudicacion_tipo = (String)jcbClaudicacion2.getSelectedItemAt();
}else{
    claudicacion_tipo = "0";
}
String claudicacion_miembro = (String)jcbClaudicacionMiembro.getSelectedItemAt();
String nervioso_ubicacion = (String)jcbNerviosoUbicacion.getSelectedItemAt();
String paralisis = (String)jcbParalisis.getSelectedItemAt();
String paralisis2 = "Si";
if(paralisis==paralisis2){

```

```

        paralis = jtfParalis.getText();
    }
    String paresia = (String)jcbParesia.getSelectedItem();
    String paresia2 = "Si";
    if(paresia == paresia2){
        paresia = jtfParesia.getText();
    }
    String convulsion = (String)jcbConvulsion.getSelectedItem();
    String convulsion2 = "Si";
    if(convulsion == convulsion2){
        convulsion = jtfConvulsion.getText();
    }
    String ataxia = (String)jcbAtaxia.getSelectedItem();
    String reflejos = jtfReflejos.getText();
    String sensibilidad = (String)jcbSensibilidad.getSelectedItem();
    String conducta = (String)jcbConducta.getSelectedItem();
    String estado_sensorio = (String)jcbEstadoSensorio.getSelectedItem();
    String snms = (String)jcbSNMS.getSelectedItem();
    String snmi = (String)jcbSNMI.getSelectedItem();
    String ojo_derecho = jtfOjoDerecho.getText();
    String ojo_izq = jtfOjoIzquierdo.getText();
    String diagnostico_presuntivo = jtaDiagnosticoPresuntivo.getText();
    String diagnostico_diferencial = jtaDiagnosticoDiferencial.getText();
    String diagnostico_pronostico = jtaDiagnosticoPronostico.getText();
    String tratamiento = jtaTratamiento.getText();
    String indicaciones = jtaIndicaciones.getText();
    String lesion_ubicacion = "dds";
    java.util.Date date = new java.util.Date();
    java.text.SimpleDateFormat sdf = new java.text.SimpleDateFormat("yyyy-MM-dd");
    String fecha = sdf.format(date);
    String datos[] =
{caso_clinico, fecha, cod_animal, dni_doctor, dni_alumno, motivo, anamnesis, temperatura, fc, fr, cc, ll_cap, pulso_ritmo, pulso_intensidad, hidratacion, ma_
ocular, ma_bucal, ma_nasal, ma_genital, submandibular, axilar, inguinales, popliteos, lesion_tipo, lesion_forma, lesion_ubicacion, lesion_simetria, olor, pruri
to, manto_piloso, ectoparasitos, oido, boca, esofago, estomago, intestino, higado, regurgitacion, vomito, diarrea, ruidos, distension, vias_superiores, vias_infe
riores, ritmo_respiratorio, tipo, tos_reflejo, auscultacion, corazon_auscultacion, ritmo_corazon, riñones, ureteres, vejiga, uretra, secreciones, orina, genitales
_interno, genitales_externo, genitales_secreciones, tacto_rectal, locomotor_lesion, locomotor_ubicacionL, locomotor_deformacion, locomotor_ubicacio
nD, claudicacion_miembro, claudicacion_tipo, nervioso_ubicacion, paralis, paresia, convulsion, ataxia, reflejos, sensibilidad, conducta, estado_sensorio, sn
ms, snmi, ojo_derecho, ojo_izq, diagnostico_presuntivo, diagnostico_diferencial, diagnostico_pronostico, tratamiento, indicaciones};
    animal.InsertarRegistro(datos, "insert into consulta
values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,
?, ?, ?)");
    }
}

private void jBuscarAlumnoActionPerformed(java.awt.event.ActionEvent evt) {
    BuscarAlumno buscarAlumno = new BuscarAlumno();
    PantallaPrincipal.jdpFondo.add(buscarAlumno);
    buscarAlumno.toFront();
    buscarAlumno.setVisible(true);
}

private void jBuscarDocenteActionPerformed(java.awt.event.ActionEvent evt) {
    BuscarDocente buscarDocente = new BuscarDocente();
    PantallaPrincipal.jdpFondo.add(buscarDocente);
    buscarDocente.toFront();
    buscarDocente.setVisible(true);
}

private void jBuscarPropietarioActionPerformed(java.awt.event.ActionEvent evt) {
    BuscarDueñoP buscar = new BuscarDueñoP();
    PantallaPrincipal.jdpFondo.add(buscar);
    buscar.toFront();
    buscar.setVisible(true);
}

private void jcbDatosDueñoActionPerformed(java.awt.event.ActionEvent evt) {
    String dni = jIDni.getText();
    String dato = jcbDatosDueño.getSelectedItem().toString();
    if(dato == "e-mail"){
        jIDato.setText(animal.GetData("email", "SELECT email FROM duenio WHERE dni_duenio = \""+dni+"\""));
    }
    if(dato == "facebook"){
        jIDato.setText(animal.GetData("facebook", "SELECT facebook FROM duenio WHERE dni_duenio = \""+dni+"\""));
    }
    if(dato == "celular"){

```

```
jldato.setText(animal.GetData("telefono_celular","SELECT telefono_celular FROM duenio WHERE dni_duenio = \"'+dni+'\""));
}
}
private void jbSalirActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}
MySQL animal = new MySQL();
// Variables declaration - do not modify
public javax.swing.JButton jBuscarAlumno;
public javax.swing.JButton jBuscarDocente;
public javax.swing.JButton jBuscarPropietario;
public javax.swing.JLabel jLabel1;
public javax.swing.JLabel jLabel10;
public javax.swing.JLabel jLabel100;
public javax.swing.JLabel jLabel101;
public javax.swing.JLabel jLabel102;
public javax.swing.JLabel jLabel103;
public javax.swing.JLabel jLabel104;
public javax.swing.JLabel jLabel105;
public javax.swing.JLabel jLabel106;
public javax.swing.JLabel jLabel107;
public javax.swing.JLabel jLabel108;
public javax.swing.JLabel jLabel109;
public javax.swing.JLabel jLabel11;
public javax.swing.JLabel jLabel110;
public javax.swing.JLabel jLabel111;
public javax.swing.JLabel jLabel112;
public javax.swing.JLabel jLabel113;
public javax.swing.JLabel jLabel114;
public javax.swing.JLabel jLabel115;
public javax.swing.JLabel jLabel116;
public javax.swing.JLabel jLabel117;
public javax.swing.JLabel jLabel118;
public javax.swing.JLabel jLabel119;
public javax.swing.JLabel jLabel12;
public javax.swing.JLabel jLabel120;
public javax.swing.JLabel jLabel121;
public javax.swing.JLabel jLabel122;
public javax.swing.JLabel jLabel123;
public javax.swing.JLabel jLabel124;
public javax.swing.JLabel jLabel125;
public javax.swing.JLabel jLabel13;
public javax.swing.JLabel jLabel14;
public javax.swing.JLabel jLabel143;
public javax.swing.JLabel jLabel15;
public static javax.swing.JLabel jLabel16;
public javax.swing.JLabel jLabel17;
public javax.swing.JLabel jLabel18;
public javax.swing.JLabel jLabel19;
public javax.swing.JLabel jLabel2;
public javax.swing.JLabel jLabel20;
public javax.swing.JLabel jLabel21;
public javax.swing.JLabel jLabel22;
public javax.swing.JLabel jLabel23;
public javax.swing.JLabel jLabel24;
public javax.swing.JLabel jLabel25;
public javax.swing.JLabel jLabel26;
public javax.swing.JLabel jLabel27;
public javax.swing.JLabel jLabel28;
public javax.swing.JLabel jLabel29;
public javax.swing.JLabel jLabel3;
public javax.swing.JLabel jLabel30;
public javax.swing.JLabel jLabel31;
public javax.swing.JLabel jLabel32;
public javax.swing.JLabel jLabel33;
public javax.swing.JLabel jLabel34;
public javax.swing.JLabel jLabel35;
public javax.swing.JLabel jLabel36;
public javax.swing.JLabel jLabel37;
public javax.swing.JLabel jLabel38;
public javax.swing.JLabel jLabel39;
```

```
public javax.swing.JLabel jLabel4;  
public javax.swing.JLabel jLabel40;  
public javax.swing.JLabel jLabel41;  
public javax.swing.JLabel jLabel42;  
public javax.swing.JLabel jLabel43;  
public javax.swing.JLabel jLabel44;  
public javax.swing.JLabel jLabel45;  
public javax.swing.JLabel jLabel46;  
public javax.swing.JLabel jLabel47;  
public javax.swing.JLabel jLabel48;  
public javax.swing.JLabel jLabel49;  
public javax.swing.JLabel jLabel5;  
public javax.swing.JLabel jLabel50;  
public javax.swing.JLabel jLabel51;  
public javax.swing.JLabel jLabel52;  
public javax.swing.JLabel jLabel53;  
public javax.swing.JLabel jLabel54;  
public javax.swing.JLabel jLabel55;  
public javax.swing.JLabel jLabel56;  
public javax.swing.JLabel jLabel57;  
public javax.swing.JLabel jLabel58;  
public javax.swing.JLabel jLabel59;  
public javax.swing.JLabel jLabel6;  
public javax.swing.JLabel jLabel60;  
public javax.swing.JLabel jLabel61;  
public javax.swing.JLabel jLabel62;  
public javax.swing.JLabel jLabel63;  
public javax.swing.JLabel jLabel64;  
public javax.swing.JLabel jLabel65;  
public javax.swing.JLabel jLabel66;  
public javax.swing.JLabel jLabel67;  
public javax.swing.JLabel jLabel68;  
public javax.swing.JLabel jLabel69;  
public javax.swing.JLabel jLabel7;  
public javax.swing.JLabel jLabel70;  
public javax.swing.JLabel jLabel71;  
public javax.swing.JLabel jLabel72;  
public javax.swing.JLabel jLabel73;  
public javax.swing.JLabel jLabel74;  
public javax.swing.JLabel jLabel75;  
public javax.swing.JLabel jLabel76;  
public javax.swing.JLabel jLabel77;  
public javax.swing.JLabel jLabel78;  
public javax.swing.JLabel jLabel79;  
public javax.swing.JLabel jLabel8;  
public javax.swing.JLabel jLabel80;  
public javax.swing.JLabel jLabel81;  
public javax.swing.JLabel jLabel82;  
public javax.swing.JLabel jLabel83;  
public javax.swing.JLabel jLabel84;  
public javax.swing.JLabel jLabel85;  
public javax.swing.JLabel jLabel86;  
public javax.swing.JLabel jLabel87;  
public javax.swing.JLabel jLabel88;  
public javax.swing.JLabel jLabel89;  
public javax.swing.JLabel jLabel9;  
public javax.swing.JLabel jLabel90;  
public javax.swing.JLabel jLabel91;  
public javax.swing.JLabel jLabel92;  
public javax.swing.JLabel jLabel93;  
public javax.swing.JLabel jLabel94;  
public javax.swing.JLabel jLabel95;  
public javax.swing.JLabel jLabel96;  
public javax.swing.JLabel jLabel97;  
public javax.swing.JLabel jLabel98;  
public javax.swing.JLabel jLabel99;  
public javax.swing.JScrollPane jScrollPane1;  
public javax.swing.JScrollPane jScrollPane10;  
public javax.swing.JScrollPane jScrollPane11;  
public javax.swing.JScrollPane jScrollPane12;  
public javax.swing.JScrollPane jScrollPane13;
```



```
public javax.swing.JScrollPane jScrollPane14;  
public javax.swing.JScrollPane jScrollPane15;  
public javax.swing.JScrollPane jScrollPane16;  
public javax.swing.JScrollPane jScrollPane17;  
public javax.swing.JScrollPane jScrollPane18;  
public javax.swing.JScrollPane jScrollPane19;  
public javax.swing.JScrollPane jScrollPane2;  
public javax.swing.JScrollPane jScrollPane20;  
public javax.swing.JScrollPane jScrollPane21;  
public javax.swing.JScrollPane jScrollPane22;  
public javax.swing.JScrollPane jScrollPane23;  
public javax.swing.JScrollPane jScrollPane24;  
public javax.swing.JScrollPane jScrollPane25;  
public javax.swing.JScrollPane jScrollPane26;  
public javax.swing.JScrollPane jScrollPane27;  
public javax.swing.JScrollPane jScrollPane28;  
public javax.swing.JScrollPane jScrollPane29;  
public javax.swing.JScrollPane jScrollPane3;  
public javax.swing.JScrollPane jScrollPane31;  
public javax.swing.JScrollPane jScrollPane32;  
public javax.swing.JScrollPane jScrollPane33;  
public javax.swing.JScrollPane jScrollPane34;  
public javax.swing.JScrollPane jScrollPane35;  
public javax.swing.JScrollPane jScrollPane36;  
public static javax.swing.JScrollPane jScrollPane37;  
public javax.swing.JScrollPane jScrollPane4;  
public javax.swing.JScrollPane jScrollPane5;  
public javax.swing.JScrollPane jScrollPane6;  
public javax.swing.JScrollPane jScrollPane7;  
public javax.swing.JScrollPane jScrollPane8;  
public javax.swing.JScrollPane jScrollPane9;  
public static javax.swing.JTabbedPane jTabbedPane1;  
public javax.swing.JButton jbGuardar;  
public javax.swing.JButton jbSalir;  
public javax.swing.JComboBox jcbAtaxia;  
public javax.swing.JComboBox jcbAxilar1;  
public javax.swing.JComboBox jcbAxilar2;  
public javax.swing.JComboBox jcbAxilar3;  
public javax.swing.JComboBox jcbAxilar4;  
public javax.swing.JComboBox jcbCC;  
public javax.swing.JComboBox jcbClaudicacion1;  
public javax.swing.JComboBox jcbClaudicacion2;  
public javax.swing.JComboBox jcbClaudicacionMiembro;  
public javax.swing.JComboBox jcbConduta;  
public javax.swing.JComboBox jcbConvulsion;  
public javax.swing.JComboBox jcbDatosDueño;  
public javax.swing.JComboBox jcbDiarrea;  
public javax.swing.JComboBox jcbDistension;  
public javax.swing.JComboBox jcbEstadoSensorio;  
public javax.swing.JComboBox jcbGenitalesSecreciones;  
public javax.swing.JComboBox jcbHidratacion;  
public javax.swing.JComboBox jcbInguinales1;  
public javax.swing.JComboBox jcbInguinales2;  
public javax.swing.JComboBox jcbInguinales3;  
public javax.swing.JComboBox jcbInguinales4;  
public javax.swing.JComboBox jcbLLCap;  
public javax.swing.JComboBox jcbLesion;  
public javax.swing.JComboBox jcbLesionEctoparasitos;  
public javax.swing.JComboBox jcbLesionPrurito;  
public javax.swing.JComboBox jcbLesionSimetria;  
public javax.swing.JComboBox jcbMaBucal;  
public javax.swing.JComboBox jcbMaGenital;  
public javax.swing.JComboBox jcbMaNasal;  
public javax.swing.JComboBox jcbMaOcular;  
public javax.swing.JComboBox jcbNerviosoUbicacion;  
public javax.swing.JComboBox jcbParalisis;  
public javax.swing.JComboBox jcbParesia;  
public javax.swing.JComboBox jcbPopliteos1;  
public javax.swing.JComboBox jcbPopliteos2;  
public javax.swing.JComboBox jcbPopliteos3;  
public javax.swing.JComboBox jcbPopliteos4;
```



```
public javax.swing.JComboBox jcbPulsoIntensidad;
public javax.swing.JComboBox jcbPulsoRitmo;
public javax.swing.JComboBox jcbRegurgitacion;
public javax.swing.JComboBox jcbRitmoCorazon;
public javax.swing.JComboBox jcbRitmoRespiratorio;
public javax.swing.JComboBox jcbRuidos;
public javax.swing.JComboBox jcbSNMI;
public javax.swing.JComboBox jcbSNMS;
public javax.swing.JComboBox jcbSecreciones;
public javax.swing.JComboBox jcbSensibilidad;
public javax.swing.JComboBox jcbSubmandibular1;
public javax.swing.JComboBox jcbSubmandibular2;
public javax.swing.JComboBox jcbSubmandibular3;
public javax.swing.JComboBox jcbSubmandibular4;
public javax.swing.JComboBox jcbTipo;
public javax.swing.JComboBox jcbTosReflejo;
public javax.swing.JComboBox jcbVomito;
public static javax.swing.JLabel jlAlumno;
public static javax.swing.JLabel jlAlumno1;
public static javax.swing.JLabel jlAlumno2;
public static javax.swing.JLabel jlApellido;
public javax.swing.JLabel jlCasoClinico;
public static javax.swing.JLabel jlColor;
public javax.swing.JLabel jlDato;
public static javax.swing.JLabel jlDni;
public static javax.swing.JLabel jlDocente;
public static javax.swing.JLabel jlDocente1;
public static javax.swing.JLabel jlDocente2;
public static javax.swing.JLabel jlDomicilio;
public static javax.swing.JLabel jlEspecie;
public javax.swing.JLabel jlFecha;
public static javax.swing.JLabel jlFechaNacimiento;
public javax.swing.JLabel jlHora;
public static javax.swing.JLabel jlNombre;
public static javax.swing.JLabel jlPeso;
public static javax.swing.JLabel jlRaza;
public static javax.swing.JLabel jlSexo;
public static javax.swing.JLabel jlTelefono;
public javax.swing.JPanel jpAxilar;
public javax.swing.JPanel jpClaudicacion;
public javax.swing.JPanel jpConsulta1;
public javax.swing.JPanel jpConsulta10;
public javax.swing.JPanel jpConsulta2;
public javax.swing.JPanel jpConsulta3;
public javax.swing.JPanel jpConsulta4;
public javax.swing.JPanel jpConsulta5;
public javax.swing.JPanel jpConsulta6;
public javax.swing.JPanel jpConsulta7;
public javax.swing.JPanel jpConsulta8;
public javax.swing.JPanel jpConsulta9;
public javax.swing.JPanel jpConvulsion;
public javax.swing.JPanel jpDeshidratado;
public static javax.swing.JPanel jpGeneral;
public javax.swing.JPanel jpGenitalesSecreciones;
public javax.swing.JPanel jpInguinales;
public javax.swing.JPanel jpLesion;
public javax.swing.JPanel jpParalisis;
public javax.swing.JPanel jpParesia;
public javax.swing.JPanel jpPopliteos;
public javax.swing.JPanel jpSecreciones;
public javax.swing.JPanel jpSubmandibular;
public javax.swing.JTextArea jtaAnamnesis;
public javax.swing.JTextArea jtaAuscultacion;
public javax.swing.JTextArea jtaBoca;
public javax.swing.JTextArea jtaCodExamenes;
public javax.swing.JTextArea jtaCorazonAuscultacion;
public javax.swing.JTextArea jtaDiagnosticoDiferencial;
public javax.swing.JTextArea jtaDiagnosticoPresuntivo;
public javax.swing.JTextArea jtaDiagnosticoPronostico;
public javax.swing.JTextArea jtaEsofago;
public javax.swing.JTextArea jtaEstomago;
```

```
public javax.swing.JTextArea jtaGenitalesExterno;
public javax.swing.JTextArea jtaGenitalesInterno;
public javax.swing.JTextArea jtaGenitalesSecreciones;
public javax.swing.JTextArea jtaHigado;
public javax.swing.JTextArea jtaIndicaciones;
public javax.swing.JTextArea jtaInstetinos;
public javax.swing.JTextArea jtaLesionManto;
public javax.swing.JTextArea jtaLesionOidos;
public javax.swing.JTextArea jtaLesionOlor;
public javax.swing.JTextArea jtaLesionTipo;
public javax.swing.JTextArea jtaLocomotorDesformacion;
public javax.swing.JTextArea jtaLocomotorLesion;
public javax.swing.JTextArea jtaLocomotorUbicacionD;
public javax.swing.JTextArea jtaLocomotorUbicacionL;
public javax.swing.JTextArea jtaMotivo;
public javax.swing.JTextArea jtaOrina;
public javax.swing.JTextArea jtaRiñones;
public javax.swing.JTextArea jtaSecreciones;
public javax.swing.JTextArea jtaTactoRectal;
public javax.swing.JTextArea jtaTratamiento;
public javax.swing.JTextArea jtaUreteres;
public javax.swing.JTextArea jtaUretra;
public javax.swing.JTextArea jtaVejiga;
public javax.swing.JTextArea jtaViasInferiores;
public javax.swing.JTextArea jtaViasSuperiores;
public javax.swing.JTextField jtfConvulsion;
public javax.swing.JTextField jtfDeshidratado;
public javax.swing.JTextField jtfFC;
public javax.swing.JTextField jtfFR;
public javax.swing.JTextField jtfLesionForma;
public javax.swing.JTextField jtfOjoDerecho;
public javax.swing.JTextField jtfOjolzquierdo;
public javax.swing.JTextField jtfParalisis;
public javax.swing.JTextField jtfParesia;
public javax.swing.JTextField jtfReflejos;
public javax.swing.JTextField jtftemperatura;
public javax.swing.JTabbedPane jtpPestañas;
// End of variables declaration
}
```

DerivacionG.Java

```

package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import java.sql.Date;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
public class DerivacionG extends javax.swing.JInternalFrame {
    public DerivacionG() {
        initComponents();
        jlCodAnimal.setVisible(false);
        Calendar Cal= Calendar.getInstance();
        int Max= Integer.parseInt(animal.GetData("cod_derivacion", "SELECT MAX(cod_derivacion) cod_derivacion FROM derivaciong"));
        JOptionPane.showMessageDialog(rootPane, Max);
        int num= Max +1;
        jlcodigoderivacion.setText(String.valueOf(num));
        String fec= Cal.get(Cal.DATE)+"/"+(Cal.get(Cal.MONTH)+1)+"/"+Cal.get(Cal.YEAR);
        String hor=Cal.get(Cal.HOUR_OF_DAY)+":"+Cal.get(Cal.MINUTE)+":"+Cal.get(Cal.SECOND);
        jlFecha.setText(fec);
        jlHora.setText(hor);
    }
    /* 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")
    private void jbGuardarActionPerformed(java.awt.event.ActionEvent evt) {
        String cod_derivacion = jlcodigoderivacion.getText();
        String dni= jlDni.getText();
        String cod_animal=jlCodAnimal.getText();
        String dni_doctor=jlDocente2.getText();
        String dni_alumno=jlAlumno2.getText();
        boolean aux = animal.existe("derivaciong", "cod_derivacion", cod_derivacion);
        if (aux==true){
            JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
        }else{
            String solicitud = jtaSolicitud.getText();
            String sistema = jtaSistema.getText();
            String sugerencia = jtaSugerencia.getText();
            java.util.Date date = new java.util.Date();
            java.text.SimpleDateFormat sdf=new java.text.SimpleDateFormat("yyyy-MM-dd");
            String fecha = sdf.format(date);
            String datos[] = {cod_derivacion,fecha,cod_animal,dni_doctor,dni_alumno,solicitud,sistema,sugerencia};
            animal.InsertarRegistro(datos, "insert into derivaciong values(?,?,?,?,?,?,?,?)");
        }
    }
    private void jBuscarAlumnoActionPerformed(java.awt.event.ActionEvent evt) {
        BuscarAlumnoDerivacionG buscarAlumno = new BuscarAlumnoDerivacionG();
        PantallaPrincipal.jdpFondo.add(buscarAlumno);
        buscarAlumno.toFront();
        buscarAlumno.setVisible(true);
    }
    private void jBuscarDocenteActionPerformed(java.awt.event.ActionEvent evt) {
        BuscarDocenteDerivacionG buscarDocente = new BuscarDocenteDerivacionG();
        PantallaPrincipal.jdpFondo.add(buscarDocente);
        buscarDocente.toFront();
        buscarDocente.setVisible(true);
    }
    private void jBuscarPropietarioActionPerformed(java.awt.event.ActionEvent evt) {
        BuscarDueñoDerivacionG buscar = new BuscarDueñoDerivacionG();
        PantallaPrincipal.jdpFondo.add(buscar);
        buscar.toFront();
        buscar.setVisible(true);
    }

```

```

}
private void jcbDatosDueñoActionPerformed(java.awt.event.ActionEvent evt) {
    String dni= jIDni.getText();
    String dato = jcbDatosDueño.getSelectedItem().toString();
    if(dato=="e-mail"){
        jIDato.setText(animal.GetData("email","SELECT email FROM duenio WHERE dni_duenio = \"'+dni+'\""));
    }
    if(dato=="facebook"){
        jIDato.setText(animal.GetData("facebook","SELECT facebook FROM duenio WHERE dni_duenio = \"'+dni+'\""));
    }
    if(dato=="celular"){
        jIDato.setText(animal.GetData("telefono_celular","SELECT telefono_celular FROM duenio WHERE dni_duenio = \"'+dni+'\""));
    }
}
MySQL animal = new MySQL();
// Variables declaration - do not modify
public javax.swing.JButton jBuscarAlumno;
public javax.swing.JButton jBuscarDocente;
public javax.swing.JButton jBuscarPropietario;
public javax.swing.JLabel jLabel1;
public javax.swing.JLabel jLabel10;
public javax.swing.JLabel jLabel11;
public javax.swing.JLabel jLabel116;
public javax.swing.JLabel jLabel12;
public javax.swing.JLabel jLabel123;
public javax.swing.JLabel jLabel124;
public javax.swing.JLabel jLabel125;
public javax.swing.JLabel jLabel126;
public javax.swing.JLabel jLabel13;
public javax.swing.JLabel jLabel14;
public javax.swing.JLabel jLabel143;
public javax.swing.JLabel jLabel15;
public static javax.swing.JLabel jLabel16;
public javax.swing.JLabel jLabel18;
public javax.swing.JLabel jLabel2;
public javax.swing.JLabel jLabel3;
public javax.swing.JLabel jLabel4;
public javax.swing.JLabel jLabel5;
public javax.swing.JLabel jLabel6;
public javax.swing.JLabel jLabel7;
public javax.swing.JLabel jLabel8;
public javax.swing.JLabel jLabel9;
public javax.swing.JScrollPane jScrollPane1;
public javax.swing.JScrollPane jScrollPane2;
public javax.swing.JScrollPane jScrollPane35;
public static javax.swing.JScrollPane jScrollPane37;
public static javax.swing.JTabbedPane jTabbedPane1;
public javax.swing.JButton jbGuardar;
public javax.swing.JButton jbSalir;
public javax.swing.JComboBox jcbDatosDueño;
public static javax.swing.JLabel jAlumno;
public static javax.swing.JLabel jAlumno1;
public static javax.swing.JLabel jAlumno2;
public static javax.swing.JLabel jApellido;
public static javax.swing.JLabel jICodAnimal;
public static javax.swing.JLabel jIColor;
public javax.swing.JLabel jIDato;
public static javax.swing.JLabel jIDni;
public static javax.swing.JLabel jIDocente;
public static javax.swing.JLabel jIDocente1;
public static javax.swing.JLabel jIDocente2;
public static javax.swing.JLabel jIDomicilio;
public static javax.swing.JLabel jIEspecie;
public javax.swing.JLabel jIFecha;
public static javax.swing.JLabel jIFechaNacimiento;
public javax.swing.JLabel jIHora;
public static javax.swing.JLabel jINombre;
public static javax.swing.JLabel jIPeso;
public static javax.swing.JLabel jIRaza;
public static javax.swing.JLabel jISexo;
public static javax.swing.JLabel jITelefono;

```



```
public javax.swing.JLabel jlCodigoDerivacion;  
public javax.swing.JPanel jpConsulta10;  
public javax.swing.JPanel jpConvulsion;  
public static javax.swing.JPanel jpGeneral;  
public javax.swing.JPanel jpParalisis;  
public javax.swing.JPanel jpParesia;  
public javax.swing.JPanel jpRetorno1;  
public javax.swing.JPanel jpRetorno3;  
public javax.swing.JTextArea jtaSistema;  
public javax.swing.JTextArea jtaSolicitud;  
public javax.swing.JTextArea jtaSugerencia;  
public javax.swing.JTabbedPane jtpPestañas;  
// End of variables declaration  
}
```

DerivacionP.Java

```

package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import java.sql.Date;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JInternalFrame {
    public DerivacionP() {
        initComponents();
        jlcodAnimal.setVisible(false);
        Calendar Cal= Calendar.getInstance();
        int Max= Integer.parseInt(animl.GetData("cod_derivacion", "SELECT MAX(cod_derivacion) cod_derivacion FROM derivacion"));
        int num= Max +1;
        jlcodigoderivacion.setText(String.valueOf(num));
        String fec= Cal.get(Cal.DATE)+"/" + (Cal.get(Cal.MONTH)+1) + "/" + Cal.get(Cal.YEAR);
        String hor=Cal.get(Cal.HOUR_OF_DAY)+":"+Cal.get(Cal.MINUTE)+":"+Cal.get(Cal.SECOND);
        jlFecha.setText(fec);
        jlHora.setText(hor);
    }
    /* 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")
    private void jbGuardarActionPerformed(java.awt.event.ActionEvent evt) {
        String cod_derivacion = jlcodigoderivacion.getText();
        String cod_animal=jlcodAnimal.getText();
        String dni_doctor=jldocente2.getText();
        String dni_alumno=jlAlumno2.getText();
        boolean aux = animal.existe("derivacion", "cod_derivacion", cod_derivacion);
        if (aux==true){
            JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
        }else{
            String anamnesis = jtaAnamnesis.getText();
            String sistema = jtaSistema.getText();
            String indicaciones = jtaIndicaciones.getText();
            java.util.Date date = new java.util.Date();
            java.text.SimpleDateFormat sdf=new java.text.SimpleDateFormat("yyyy-MM-dd");
            String fecha = sdf.format(date);
            String datos[] = {fecha,cod_animal,dni_doctor,dni_alumno,anamnesis,sistema,indicaciones};
            animal.InsertarRegistro(datos, "insert into derivacion
            (`fecha`,`cod_animal`,`dni_doctor`,`dni_alumno`,`solicitud`,`sistema`,`sugerencia`)values(?,?,?,?,?,?,?)");
        }
    }
    private void jBuscarAlumnoActionPerformed(java.awt.event.ActionEvent evt) {
        BuscarAlumnoDerivacion buscarAlumno = new BuscarAlumnoDerivacion();
        PantallaPrincipal.jdpFondo.add(buscarAlumno);
        buscarAlumnoToFront();
        buscarAlumno.setVisible(true);
    }
    private void jBuscarDocenteActionPerformed(java.awt.event.ActionEvent evt) {
        BuscarDocenteDerivacion buscarDocente = new BuscarDocenteDerivacion();
        PantallaPrincipal.jdpFondo.add(buscarDocente);
        buscarDocenteToFront();
        buscarDocente.setVisible(true);
    }
    private void jBuscarPropietarioActionPerformed(java.awt.event.ActionEvent evt) {
        BuscarDueñoDerivacion buscar = new BuscarDueñoDerivacion();
        PantallaPrincipal.jdpFondo.add(buscar);
    }

```



```

        buscar.toFront();
        buscar.setVisible(true);
    }
    private void jcbDatosDueñoActionPerformed(java.awt.event.ActionEvent evt) {
        String dni= jIDni.getText();
        String dato = jcbDatosDueño.getSelectedItem().toString();
        if(dato=="e-mail"){
            jIDato.setText(animal.GetData("email","SELECT email FROM duenio WHERE dni_duenio = '"+dni+"'"));
        }
        if(dato=="facebook"){
            jIDato.setText(animal.GetData("facebook","SELECT facebook FROM duenio WHERE dni_duenio = '"+dni+"'"));
        }
        if(dato=="celular"){
            jIDato.setText(animal.GetData("telefono_celular","SELECT telefono_celular FROM duenio WHERE dni_duenio = '"+dni+"'"));
        }
    }
    private void jbSalirActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    MySQL animal = new MySQL();
    // Variables declaration - do not modify
    public javax.swing.JButton jBuscarAlumno;
    public javax.swing.JButton jBuscarDocente;
    public javax.swing.JButton jBuscarPropietario;
    public javax.swing.JLabel jLabel1;
    public javax.swing.JLabel jLabel10;
    public javax.swing.JLabel jLabel11;
    public javax.swing.JLabel jLabel116;
    public javax.swing.JLabel jLabel12;
    public javax.swing.JLabel jLabel123;
    public javax.swing.JLabel jLabel124;
    public javax.swing.JLabel jLabel125;
    public javax.swing.JLabel jLabel126;
    public javax.swing.JLabel jLabel13;
    public javax.swing.JLabel jLabel14;
    public javax.swing.JLabel jLabel143;
    public javax.swing.JLabel jLabel15;
    public static javax.swing.JLabel jLabel16;
    public javax.swing.JLabel jLabel18;
    public javax.swing.JLabel jLabel2;
    public javax.swing.JLabel jLabel3;
    public javax.swing.JLabel jLabel4;
    public javax.swing.JLabel jLabel5;
    public javax.swing.JLabel jLabel6;
    public javax.swing.JLabel jLabel7;
    public javax.swing.JLabel jLabel8;
    public javax.swing.JLabel jLabel9;
    public javax.swing.JScrollPane jScrollPane1;
    public javax.swing.JScrollPane jScrollPane2;
    public javax.swing.JScrollPane jScrollPane35;
    public static javax.swing.JScrollPane jScrollPane37;
    public static javax.swing.JTabbedPane jTabbedPane1;
    public javax.swing.JButton jbGuardar;
    public javax.swing.JButton jbSalir;
    public javax.swing.JComboBox jcbDatosDueño;
    public static javax.swing.JLabel jAlumno;
    public static javax.swing.JLabel jAlumno1;
    public static javax.swing.JLabel jAlumno2;
    public static javax.swing.JLabel jApellido;
    public static javax.swing.JLabel jICodAnimal;
    public static javax.swing.JLabel jIColor;
    public javax.swing.JLabel jIDato;
    public static javax.swing.JLabel jIDni;
    public static javax.swing.JLabel jIDocente;
    public static javax.swing.JLabel jIDocente1;
    public static javax.swing.JLabel jIDocente2;
    public static javax.swing.JLabel jIDomicilio;
    public static javax.swing.JLabel jIEspecie;
    public javax.swing.JLabel jIFecha;
    public static javax.swing.JLabel jIFechaNacimiento;
    public javax.swing.JLabel jIHora;

```




```
public static javax.swing.JLabel jlNombre;  
public static javax.swing.JLabel jlPeso;  
public static javax.swing.JLabel jlRaza;  
public static javax.swing.JLabel jlSexo;  
public static javax.swing.JLabel jlTelefono;  
public javax.swing.JLabel jlcodigoderivacion;  
public javax.swing.JPanel jpConsulta10;  
public javax.swing.JPanel jpConvulsion;  
public static javax.swing.JPanel jpGeneral;  
public javax.swing.JPanel jpParalisis;  
public javax.swing.JPanel jpParesia;  
public javax.swing.JPanel jpRetorno1;  
public javax.swing.JPanel jpRetorno3;  
public javax.swing.JTextArea jtaAnamnesis;  
public javax.swing.JTextArea jtaIndicaciones;  
public javax.swing.JTextArea jtaSistema;  
public javax.swing.JTabbedPane jtpPestañas;  
// End of variables declaration  
}
```

DetalleP.Java

```
package Consultas;
import ModuloAnimal.*;
public class DetalleP extends javax.swing.JInternalFrame {
    public DetalleP() {
        initComponents();
        int fila = BuscarDueñoP.jtAnimal.getSelectedRow();
        jlApellido.setText(BuscarDueñoP.jtAnimal.getValueAt(fila, 0).toString());
        jlNombre.setText(BuscarDueñoP.jtAnimal.getValueAt(fila, 1).toString());
        jlDni.setText(BuscarDueñoP.jtAnimal.getValueAt(fila, 2).toString());
        jlNombreAnimal.setText(BuscarDueñoP.jtAnimal.getValueAt(fila, 3).toString());
        jlEspecie.setText(BuscarDueñoP.jtAnimal.getValueAt(fila, 4).toString());
        jlRaza.setText(BuscarDueñoP.jtAnimal.getValueAt(fila, 5).toString());
        jlPeso.setText(BuscarDueñoP.jtAnimal.getValueAt(fila, 6).toString());
        jlFecha.setText(BuscarDueñoP.jtAnimal.getValueAt(fila, 7).toString());
        jlSexo.setText(BuscarDueñoP.jtAnimal.getValueAt(fila, 8).toString());
        jlColor.setText(BuscarDueñoP.jtAnimal.getValueAt(fila, 9).toString());
        jlVivo.setText(BuscarDueñoP.jtAnimal.getValueAt(fila, 10).toString());
    }
    /* 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")
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    // Variables declaration - do not modify
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JButton jbCancelar;
    private javax.swing.JLabel jlApellido;
    private javax.swing.JLabel jlColor;
    private javax.swing.JLabel jlDni;
    private javax.swing.JLabel jlEspecie;
    private javax.swing.JLabel jlFecha;
    private javax.swing.JLabel jlNombre;
    private javax.swing.JLabel jlNombreAnimal;
    private javax.swing.JLabel jlPeso;
    private javax.swing.JLabel jlRaza;
    private javax.swing.JLabel jlSexo;
    private javax.swing.JLabel jlVivo;
    // End of variables declaration
}
```

ListadoDerivacionG.Java

```
package Consultas;
import Clases.MySQL;
import static Clases.PantallaPrincipal.jdpFondo;
import javax.swing.table.DefaultTableModel;
public class ListadoDerivacionG extends javax.swing.JInternalFrame {
    public ListadoDerivacionG() {
        initComponents();
        auxenviar.setVisible(false);
        auxenviar2.setVisible(false);
    }
    /* 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")
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String texto= jtfbuscar.getText();
        String[] columnNames = {"Cod. Derivacion","Fecha","Cod. Animal","Dni Doctor","Dni Alumno","solicitud","sistema","sugerencia","Nombre","Dni Dueño"};
        String[] colName = {"cod_derivacion","fecha","cod_animal","dni_doctor","dni_alumno","solicitud","sistema","sugerencia","nombre","dni_duenio"};
        //nombre de columnas, tabla, instruccion sql
        if(jrbFecha.isSelected()==true){
            dtDerivacion = derivacion.GetTabla(colName, "animal_grandes,derivacion", "SELECT `derivacion`.`*`,`animal_grandes`.`*` FROM animal_grandes,derivacion WHERE `animal_grandes`.`cod_animal` = `derivacion`.`cod_animal` AND `derivacion`.`fecha` = '"+texto+"'");
        }
        if(jrbDniDueño.isSelected()==true){
            dtDerivacion = derivacion.GetTabla(colName, "animal_grandes,derivacion", "SELECT `derivacion`.`*`,`animal_grandes`.`*` FROM animal_grandes,derivacion WHERE (`animal_grandes`.`cod_animal` = `derivacion`.`cod_animal`) AND (`animal_grandes`.`dni_duenio` = '"+texto+"'");
        }
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtDerivacion,columnNames);
        jtDuenio.setModel(datos);
        int a = jtDuenio.getRowCount();
        //depurador de jtDuenio de item null
        int i=0;
        a--;
        while(a>=i){
            if(jtDuenio.getValueAt(i,1)==null){
                ((DefaultTableModel)jtDuenio.getModel()).removeRow(i);
                a--;
                i--;
            }
            i++;
        }
    }
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar= jtDuenio.getSelectedRow();
        if(filamodificar>=0){
            String cc= jtDuenio.getValueAt(filamodificar, 2).toString();
            String r= jtDuenio.getValueAt(filamodificar, 0).toString();
            auxenviar.setText(cc);
            auxenviar2.setText(r);
        }
        this.dispose();
        MostrarDerivacionAnimalG retorno = new MostrarDerivacionAnimalG();
        jdpFondo.add(retorno);
        retorno.show();
    }
    Object[][] dtDerivacion;
    MySQL derivacion = new MySQL();
    // Variables declaration - do not modify
    private javax.swing.JLabel auxenviar;
    private javax.swing.JLabel auxenviar2;
    private javax.swing.JPanel jPanel2;
```



```
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JButton jbBuscar;  
private javax.swing.JButton jbElegir;  
private javax.swing.JRadioButton jrbDniDueño;  
private javax.swing.JRadioButton jrbFecha;  
private javax.swing.JTable jtDuenio;  
private javax.swing.JTextField jtfbuscar;  
// End of variables declaration  
}
```

ListadoDerivacionP.Java

```
package Consultas;
import Clases.MySQL;
import static Clases.PantallaPrincipal.jdpFondo;
import javax.swing.table.DefaultTableModel;
public class ListadoDerivacionP extends javax.swing.JInternalFrame {
    public ListadoDerivacionP() {
        initComponents();
        auxenviar.setVisible(false);
        auxenviar2.setVisible(false);
    }
    /* 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")
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String texto= jtfbuscar.getText();
        String[] columNames = {"Cod. Derivacion", "Fecha", "Cod. Animal", "Dni Doctor", "Dni Alumno", "solicitud", "sistema", "sugerencia", "Nombre", "Dni Dueño"};
        String[] colName = {"cod_derivacion", "fecha", "cod_animal", "dni_doctor", "dni_alumno", "solicitud", "sistema", "sugerencia", "nombre", "dni_duenio"};
        //nombre de columnas, tabla, instruccion sql
        if(jrbFecha.isSelected()==true){
            dtDerivacion = derivacion.GetTabla(colName, "animal_pequenos,derivacion", "SELECT `derivacion`.*,`animal_pequenos`.* FROM animal_pequenos,derivacion WHERE `animal_pequenos`.`cod_animal` = `derivacion`.`cod_animal` AND `derivacion`.`fecha`='"+texto+"'");
        }
        if(jrbDniDueño.isSelected()==true){
            dtDerivacion = derivacion.GetTabla(colName, "animal_pequenos,derivacion", "SELECT `derivacion`.*,`animal_pequenos`.* FROM animal_pequenos,derivacion WHERE (`animal_pequenos`.`cod_animal` = `derivacion`.`cod_animal`) AND (`animal_pequenos`.`dni_duenio`='"+texto+"'");
        }
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtDerivacion,columNames);
        jtDuenio.setModel(datos);
        int a = jtDuenio.getRowCount();
        //depurador de jtDuenio de item null
        int i=0;
        a--;
        while(a>=i){
            if(jtDuenio.getValueAt(i,1)==null){
                ((DefaultTableModel)jtDuenio.getModel()).removeRow(i);
                a--;
                i--;
            }
            i++;
        }
    }
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar= jtDuenio.getSelectedRow();
        if(filamodificar>=0){
            String cc= jtDuenio.getValueAt(filamodificar, 2).toString();
            String r= jtDuenio.getValueAt(filamodificar, 0).toString();
            auxenviar.setText(cc);
            auxenviar2.setText(r);
        }
        this.dispose();
        MostrarDerivacionAnimalP retorno = new MostrarDerivacionAnimalP();
        jdpFondo.add(retorno);
    }
}
```

```
        retorno.show();
    }
Object[][] dtDerivacion;
MySQL derivacion = new MySQL();
// Variables declaration - do not modify
public static javax.swing.JLabel auxenviar;
public static javax.swing.JLabel auxenviar2;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JButton jbBuscar;
private javax.swing.JButton jbElegir;
private javax.swing.JRadioButton jrbDniDueño;
private javax.swing.JRadioButton jrbFecha;
private javax.swing.JTable jtDuenio;
private javax.swing.JTextField jtfbuscar;
// End of variables declaration
}
```

ListadoMostrarRetornoAG.Java

```

package Consultas;
import Clases.MySQL;
import static Clases.PantallaPrincipal.jdpFondo;
import javax.swing.table.DefaultTableModel;
public class ListadoMostrarRetornoAG extends javax.swing.JInternalFrame {
    public ListadoMostrarRetornoAG() {
        initComponents();
        auxenviar.setVisible(false);
        auxenviar2.setVisible(false);
        String cc = BuscarCasoClinicoMostrarRetornoAG.auxenviar.getText();
        String[] columnNames = {"Cod. Retorno","Caso Clinico","Fecha","Cod. Animal","Nombre","Dni Dueño","apellido","nombre","Dni
        Doctor","apellido","nombre","Dni Alumno","apellido","nombre"};
        String[] colName =
{"cod_retorno","caso_clinico","fecha","cod_animal","animal_grandes.nombre","animal_grandes.dni_duenio","duenio.apellido","duenio.nombre","re
        tornog.dni_doctor","doctor.apellido","doctor.nombre","retornog.dni_alumno","alumno.apellido","alumno.nombre"};
        //nombre de columnas, tabla, instruccion sql
        dtRetorno = retorno.GetTabla(colName, "animal_grandes,duenio,doctor,alumno,retornog", "SELECT
        `retornog`.`cod_retorno`,`retornog`.`caso_clinico`,`retornog`.`fecha`,`animal_grandes`.`cod_animal`,`animal_grandes`.`nombre`,`animal_grandes`.`dn
        i_duenio`,`duenio`.`apellido`,`duenio`.`nombre`,`retornog`.`dni_doctor`,`doctor`.`apellido`,`doctor`.`nombre`,`retornog`.`dni_alumno`,`alumno`.`apelli
        do`,`alumno`.`nombre` FROM animal_grandes,duenio,doctor,alumno,retornog WHERE (`animal_grandes`.`cod_animal` = `retornog`.`cod_animal`)
        AND (`animal_grandes`.`dni_duenio` = `duenio`.`dni_duenio`) AND (`retornog`.`dni_doctor` = `doctor`.`dni_doctor`) AND (`retornog`.`dni_alumno`
        = `alumno`.`dni_alumno`) AND `retornog`.`caso_clinico`="+cc+" "; // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtRetorno,columnNames);
        jtRetorno.setModel(datos);
        int a = jtRetorno.getRowCount();
        //depurador de jtDuenio de item null
        int i=0;
        a--;
        while(a>=i){
            if(jtRetorno.getValueAt(i,1)==null){
                ((DefaultTableModel)jtRetorno.getModel()).removeRow(i);
                a--;
                i--;
            }
            i++;
        }
    }
    /* 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")
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar= jtRetorno.getSelectedRow();
        if(filamodificar>=0){
            String cc= jtRetorno.getValueAt(filamodificar, 1).toString();
            String r= jtRetorno.getValueAt(filamodificar, 0).toString();
            auxenviar.setText(cc);
            auxenviar2.setText(r);
        }
        this.dispose();
        MostrarRetornoAnimalG retorno = new MostrarRetornoAnimalG();
        jdpFondo.add(retorno);
        retorno.show();
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
}
Object[][] dtRetorno;
```




```
MySQL retorno = new MySQL();  
// Variables declaration - do not modify  
public static javax.swing.JLabel auxenviar;  
public static javax.swing.JLabel auxenviar2;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JButton jbCancelar;  
private javax.swing.JButton jbElegir;  
private javax.swing.JTable jtRetorno;  
// End of variables declaration  
}
```

ListadoMostrarRetornoAP.Java

```
package Consultas;
import Clases.MySQL;
import static Clases.PantallaPrincipal.jdpFondo;
import javax.swing.table.DefaultTableModel;
public class ListadoMostrarRetornoAP extends javax.swing.JInternalFrame {
    public ListadoMostrarRetornoAP() {
        initComponents();
        auxenviar.setVisible(false);
        auxenviar2.setVisible(false);
        String cc = BuscarCasoClinicoMostrarRetornoAP.auxenviar.getText();
        String[] columnNames = {"Cod. Retorno", "Caso Clinico", "Fecha", "Cod. Animal", "Nombre", "Dni Dueño", "apellido", "nombre", "Dni
        Doctor", "apellido", "nombre", "Dni Alumno", "apellido", "nombre"};
        String[] colName =
{"cod_retorno", "caso_clinico", "fecha", "cod_animal", "animal_pequenos.nombre", "animal_pequenos.dni_duenio", "duenio.apellido", "duenio.nombre",
"retorno.dni_doctor", "doctor.apellido", "doctor.nombre", "retorno.dni_alumno", "alumno.apellido", "alumno.nombre"};
        //nombre de columnas, tabla, instruccion sql
        dtRetorno = retorno.GetTabla(colName, "animal_pequenos,duenio,doctor,alumno,retorno", "SELECT
`retorno`.`cod_retorno`,`retorno`.`caso_clinico`,`retorno`.`fecha`,`animal_pequenos`.`cod_animal`,`animal_pequenos`.`nombre`,`animal_pequenos`.`
dni_duenio`,`duenio`.`apellido`,`duenio`.`nombre`,`retorno`.`dni_doctor`,`doctor`.`apellido`,`doctor`.`nombre`,`retorno`.`dni_alumno`,`alumno`.`ape
llido`,`alumno`.`nombre` FROM animal_pequenos,duenio,doctor,alumno,retorno WHERE (`animal_pequenos`.`cod_animal`=`retorno`.`cod_animal`)
AND (`animal_pequenos`.`dni_duenio`=`duenio`.`dni_duenio`) AND (`retorno`.`dni_doctor`=`doctor`.`dni_doctor`) AND (`retorno`.`dni_alumno`
=`alumno`.`dni_alumno`) AND `retorno`.`caso_clinico`='"+cc+"'");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtRetorno,columnNames);
        jtRetorno.setModel(datos);
        int a = jtRetorno.getRowCount();
        //depurador de jtDuenio de item null
        int i=0;
        a--;
        while(a>=i){
            if(jtRetorno.getValueAt(i,1)==null){
                ((DefaultTableModel)jtRetorno.getModel()).removeRow(i);
                a--;
                i--;
            }
            i++;
        }
    }
    /* 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")
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar= jtRetorno.getSelectedRow();
        if(filamodificar>=0){
            String cc= jtRetorno.getValueAt(filamodificar, 1).toString();
            String r= jtRetorno.getValueAt(filamodificar, 0).toString();
            auxenviar.setText(cc);
            auxenviar2.setText(r);
        }
        this.dispose();
        MostrarRetornoAnimalP retorno = new MostrarRetornoAnimalP();
        jdpFondo.add(retorno);
        retorno.show();
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
}
Object[][] dtRetorno;
MySQL retorno = new MySQL();
// Variables declaration - do not modify
public static javax.swing.JLabel auxenviar;
```



```
public static javax.swing.JLabel auxenviar2;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JButton jbCancelar;  
private javax.swing.JButton jbElegir;  
private javax.swing.JTable jtRetorno;  
// End of variables declaration  
}
```

MostrarConsultaAnimalG.Java

```

package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import java.sql.Date;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
public class MostrarConsultaAnimalG extends javax.swing.JInternalFrame {
    public MostrarConsultaAnimalG() {
        initComponents();
        String cc = BuscarMostrarCasoClinicoG.auxenviar.getText();
        String fecha1 = BuscarMostrarCasoClinicoG.auxenviar1.getText();
        Date fecha = Date.valueOf(animal.GetData("fecha", "SELECT fecha FROM consultag WHERE caso_clinico = \"\"+cc+\"\""));
        jlFecha.setText(fecha.toString());
        jlCasoClinico.setText("\""+BuscarMostrarCasoClinicoG.auxenviar.getText()+"\"/"+fecha1);
        int cod_animal = Integer.parseInt(animal.GetData("cod_animal", "SELECT cod_animal FROM consultag WHERE caso_clinico = \"\"+cc+\"\""));
        jlNombre.setText(animal.GetData("nombre", "SELECT nombre FROM animal_grandes WHERE cod_animal = \"\"+cod_animal+\"\""));
        int especie = Integer.parseInt(animal.GetData("cod_especie", "SELECT cod_especie FROM animal_grandes WHERE cod_animal = \"\"+cod_animal+\"\""));
        jlEspecie.setText(animal.GetData("especie", "SELECT especie FROM lista_especies WHERE cod_especie = \"\"+especie+\"\""));
        int raza = Integer.parseInt(animal.GetData("cod_raza", "SELECT cod_raza FROM animal_grandes WHERE cod_animal = \"\"+cod_animal+\"\""));
        jlRaza.setText(animal.GetData("raza", "SELECT raza FROM lista_razas WHERE cod_raza = \"\"+raza+\"\""));
        jlPeso.setText(animal.GetData("peso", "SELECT peso FROM animal_grandes WHERE cod_animal = \"\"+cod_animal+\"\""));
        jlFechaNacimiento.setText(animal.GetData("nacimiento", "SELECT nacimiento FROM animal_grandes WHERE cod_animal = \"\"+cod_animal+\"\""));
        jlSexo.setText(animal.GetData("sexo", "SELECT sexo FROM animal_grandes WHERE cod_animal = \"\"+cod_animal+\"\""));
        jlColor.setText(animal.GetData("color", "SELECT color FROM animal_grandes WHERE cod_animal = \"\"+cod_animal+\"\""));
        int dueño = Integer.parseInt(animal.GetData("dni_duenio", "SELECT dni_duenio FROM animal_grandes WHERE cod_animal = \"\"+cod_animal+\"\""));
        jlDni.setText(animal.GetData("dni_duenio", "SELECT dni_duenio FROM animal_grandes WHERE cod_animal = \"\"+cod_animal+\"\""));
        jlApellido.setText(animal.GetData("apellido", "SELECT apellido FROM duenio WHERE dni_duenio = \"\"+dueño+\"\""));
        jlDomicilio.setText(animal.GetData("domicilio", "SELECT domicilio FROM duenio WHERE dni_duenio = \"\"+dueño+\"\""));
        jlTelefono.setText(animal.GetData("telefono_fijo", "SELECT telefono_fijo FROM duenio WHERE dni_duenio = \"\"+dueño+\"\""));
        jlDato.setText(animal.GetData("email", "SELECT email FROM duenio WHERE dni_duenio = \"\"+dueño+\"\""));
        int dni_doctor = Integer.parseInt(animal.GetData("dni_doctor", "SELECT dni_doctor FROM consultag WHERE caso_clinico = \"\"+cc+\"\""));
        jlDocente2.setText(String.valueOf(dni_doctor));
        jlDocente.setText(animal.GetData("apellido", "SELECT apellido FROM doctor WHERE dni_doctor = \"\"+dni_doctor+\"\""));
        jlDocente1.setText(animal.GetData("nombre", "SELECT nombre FROM doctor WHERE dni_doctor = \"\"+dni_doctor+\"\""));
        int dni_alumno = Integer.parseInt(animal.GetData("dni_alumno", "SELECT dni_alumno FROM consultag WHERE caso_clinico = \"\"+cc+\"\""));
        jlAlumno2.setText(String.valueOf(dni_alumno));
        jlAlumno.setText(animal.GetData("apellido", "SELECT apellido FROM alumno WHERE dni_alumno = \"\"+dni_alumno+\"\""));
        jlAlumno1.setText(animal.GetData("nombre", "SELECT nombre FROM alumno WHERE dni_alumno = \"\"+dni_alumno+\"\""));
        jtaMotivo.setText(animal.GetData("motivo", "SELECT motivo FROM consultag WHERE caso_clinico = \"\"+cc+\"\""));
        jtaAnamnesis.setText(animal.GetData("anamnesis", "SELECT anamnesis FROM consultag WHERE caso_clinico = \"\"+cc+\"\""));
        jtfTemperatura.setText(animal.GetData("temperatura", "SELECT temperatura FROM consultag WHERE caso_clinico = \"\"+cc+\"\""));
        jtfFC.setText(animal.GetData("fc", "SELECT fc FROM consultag WHERE caso_clinico = \"\"+cc+\"\""));
        jtfFR.setText(animal.GetData("fr", "SELECT fr FROM consultag WHERE caso_clinico = \"\"+cc+\"\""));
        jcbCC.setSelectedItem(animal.GetData("cc", "SELECT cc FROM consultag WHERE caso_clinico = \"\"+cc+\"\""));
        jcbLLCap.setSelectedItem(animal.GetData("ll.cap", "SELECT ll.cap FROM consultag WHERE caso_clinico = \"\"+cc+\"\""));
        jcbPulsoIntensidad.setSelectedItem(animal.GetData("pulso_intensidad", "SELECT pulso_intensidad FROM consultag WHERE caso_clinico = \"\"+cc+\"\""));
        jcbPulsoRitmo.setSelectedItem(animal.GetData("pulso_Ritmo", "SELECT pulso_Ritmo FROM consultag WHERE caso_clinico = \"\"+cc+\"\""));
    }
}

```

```

String hidratacion2="Normal";
String hidratacion=(animal.GetData("hidratacion", "SELECT hidratacion FROM consultag WHERE caso_clinico = \"'+cc+'\""));
if (hidratacion2.equals(hidratacion)){
    jcbHidratacion.setSelectedItem(hidratacion);
}else{
    jtfDeshidratado.setText(hidratacion);
    jcbHidratacion.setSelectedItem("Deshidratado");
}
jcbMaOcular.setSelectedItem(animal.GetData("ma_ocular", "SELECT ma_ocular FROM consultag WHERE caso_clinico = \"'+cc+'\""));
jcbMaBucal.setSelectedItem(animal.GetData("ma_bucal", "SELECT ma_bucal FROM consultag WHERE caso_clinico = \"'+cc+'\""));
jcbMaNasal.setSelectedItem(animal.GetData("ma_nasal", "SELECT ma_nasal FROM consultag WHERE caso_clinico = \"'+cc+'\""));
jcbMaGenital.setSelectedItem(animal.GetData("ma_genital", "SELECT ma_genital FROM consultag WHERE caso_clinico = \"'+cc+'\""));
String submandibular = (animal.GetData("submandibular", "SELECT submandibular FROM consultag WHERE caso_clinico = \"'+cc+'\""));
String submandibular2 = "Si";
String delimitadores = ",";
String[] palabrasSeparadas = submandibular.split(delimitadores);
if(submandibular2.equals(submandibular)){
}else{
    jcbSubmandibular1.setSelectedItem("No");
    jcbSubmandibular2.setSelectedItem(palabrasSeparadas[0]);
    jcbSubmandibular3.setSelectedItem(palabrasSeparadas[1]);
    jcbSubmandibular4.setSelectedItem(palabrasSeparadas[2]);
}
String axilar = (animal.GetData("axilar", "SELECT axilar FROM consultag WHERE caso_clinico = \"'+cc+'\""));
String axilar2 = "Si";
if(axilar.equals(axilar2)){
}else{
    jcbAxilar1.setSelectedItem("No");
    palabrasSeparadas = axilar.split(delimitadores);
    jcbAxilar2.setSelectedItem(palabrasSeparadas[0]);
    jcbAxilar3.setSelectedItem(palabrasSeparadas[1]);
    jcbAxilar4.setSelectedItem(palabrasSeparadas[2]);
}
String inguinales = (animal.GetData("inguinales", "SELECT inguinales FROM consultag WHERE caso_clinico = \"'+cc+'\""));
String inguinales2 = "Si";
if(inguinales.equals(inguinales2)){
}else{
    jcbSubmandibular1.setSelectedItem("No");
    palabrasSeparadas = inguinales.split(delimitadores);
    jcbInguinales2.setSelectedItem(palabrasSeparadas[0]);
    jcbInguinales3.setSelectedItem(palabrasSeparadas[1]);
    jcbInguinales4.setSelectedItem(palabrasSeparadas[2]);
}
String popliteos = (animal.GetData("popliteos", "SELECT popliteos FROM consultag WHERE caso_clinico = \"'+cc+'\""));
String popliteos2 = "Si";
if(popliteos.equals(popliteos2)){
}else{
    jcbPopliteos1.setSelectedItem("No");
    palabrasSeparadas = popliteos.split(delimitadores);
    jcbPopliteos2.setSelectedItem(palabrasSeparadas[0]);
    jcbPopliteos3.setSelectedItem(palabrasSeparadas[1]);
    jcbPopliteos4.setSelectedItem(palabrasSeparadas[2]);
}
jtaHallazgos.setText(animal.GetData("hallazgos", "SELECT hallazgos FROM consultag WHERE caso_clinico = \"'+cc+'\""));
jtaDiagnosticoPresuntivo.setText(animal.GetData("diagnostico_presuntivo", "SELECT diagnostico_presuntivo FROM consultag WHERE caso_clinico = \"'+cc+'\""));
jtaDiagnosticoDiferencial.setText(animal.GetData("diagnostico_diferencial", "SELECT diagnostico_diferencial FROM consultag WHERE caso_clinico = \"'+cc+'\""));
jtaDiagnosticoPronostico.setText(animal.GetData("diagnostico_pronostico", "SELECT diagnostico_pronostico FROM consultag WHERE caso_clinico = \"'+cc+'\""));
jtaTratamiento.setText(animal.GetData("tratamiento", "SELECT tratamiento FROM consultag WHERE caso_clinico = \"'+cc+'\""));
jtaIndicaciones.setText(animal.GetData("indicaciones", "SELECT indicaciones FROM consultag WHERE caso_clinico = \"'+cc+'\""));

```

```

        jtaCodExamenenes.setText("");
    }
    /*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")
    private void jbGuardarActionPerformed(java.awt.event.ActionEvent evt) {
        String caso_clinico = jlcasoClinico.getText();
        String cod_animal="1";
        String dni_doctor=jldocente2.getText();
        String dni_alumno=jlalumno2.getText();
        boolean aux = animal.existe("consultag", "caso_clinico", caso_clinico);
        if (aux==true){
            JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
        }else{
            String motivo = jtaMotivo.getText();
            String anamnesis = jtaAnamnesis.getText();
            String temperatura = jtfTemperatura.getText();
            String fc = jtfFC.getText();
            String fr = jtfFR.getText();
            String cc = (String)jcbCC.getSelectedItemAt();
            String ll_cap = (String)jcbLLCap.getSelectedItemAt();
            String pulso_ritmo = (String)jcbPulsoRitmo.getSelectedItemAt();
            String pulso_intensidad = (String)jcbPulsoIntensidad.getSelectedItemAt();
            String hidratacion2="Deshidratado";
            String hidratacion=(String)jcbHidratacion.getSelectedItemAt();
            if (hidratacion2==hidratacion){
                hidratacion = jtfDeshidratado.getText();
            }else{
                hidratacion=(String)jcbHidratacion.getSelectedItemAt();
            }
            String ma_ocular = (String)jcbMaOcular.getSelectedItemAt();
            String ma_bucal = (String)jcbMaBucal.getSelectedItemAt();
            String ma_nasal = (String)jcbMaNasal.getSelectedItemAt();
            String ma_genital = (String)jcbMaGenital.getSelectedItemAt();
            String submandibular = (String)jcbSubmandibular1.getSelectedItemAt();
            String submandibular2 = "No";
            if(submandibular==submandibular2){
                submandibular =
                (String)jcbSubmandibular2.getSelectedItemAt()+" "+jcbSubmandibular3.getSelectedItemAt()+" "+jcbSubmandibular4.getSelectedItemAt();
            }else{
                submandibular = "Si";
            }
            String axilar = (String)jcbAxilar1.getSelectedItemAt();
            String axilar2 = "No";
            if(axilar==axilar2){
                axilar = (String)jcbAxilar2.getSelectedItemAt()+" "+jcbAxilar3.getSelectedItemAt()+" "+jcbAxilar4.getSelectedItemAt();
            }else{
                axilar = "Si";
            }
            String inguinales = (String)jcbInguinales1.getSelectedItemAt();
            String inguinales2 = "No";
            if(inguinales==inguinales2){
                inguinales = (String)jcbInguinales2.getSelectedItemAt()+" "+jcbInguinales3.getSelectedItemAt()+" "+jcbInguinales4.getSelectedItemAt();
            }else{
                inguinales = "Si";
            }
            String popliteos = (String)jcbPopliteos1.getSelectedItemAt();
            String popliteos2 = "No";
            if(popliteos==popliteos2){
                popliteos = (String)jcbPopliteos2.getSelectedItemAt()+" "+jcbPopliteos3.getSelectedItemAt()+" "+jcbPopliteos4.getSelectedItemAt();
            }else{

```

```

        popliteos = "Si";
    }
    String hallazgos = jtaHallazgos.getText();
    String diagnostico_presuntivo = jtaDiagnosticoPresuntivo.getText();
    String diagnostico_diferencial = jtaDiagnosticoDiferencial.getText();
    String diagnostico_pronostico = jtaDiagnosticoPronostico.getText();
    String tratamiento = jtaTratamiento.getText();
    String indicaciones = jtaIndicaciones.getText();
    String lesion_ubicacion="dds";
    java.util.Date date = new java.util.Date();
    java.text.SimpleDateFormat sdf=new java.text.SimpleDateFormat("yyyy-MM-dd");
    String fecha = sdf.format(date);
    String datos[] =
{caso_clinico,fecha,cod_animal,dni_doctor,dni_alumno,motivo,anamnesis,temperatura,fc,fr,cc,ll_cap,pulso_ritmo,pulso_intensidad,hidratacion,ma_
ocular,ma_bucal,ma_nasal,ma_genital,submandibular,axilar,inguinales,popliteos,hallazgos,diagnostico_presuntivo,diagnostico_diferencial,diagnostico_pronostico,tratamiento,indicaciones};
    animal.InsertarRegistro(datos, "insert into consultag values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");
}

}

private void jcbDatosDueñoActionPerformed(java.awt.event.ActionEvent evt) {
    String dni= jIDni.getText();
    String dato = jcbDatosDueño.getSelectedItem().toString();
    if(dato=="e-mail"){
        jIDato.setText(animal.GetData("email","SELECT email FROM duenio WHERE dni_duenio = '"+dni+"'"));
    }
    if(dato=="facebook"){
        jIDato.setText(animal.GetData("facebook","SELECT facebook FROM duenio WHERE dni_duenio = '"+dni+"'"));
    }
    if(dato=="celular"){
        jIDato.setText(animal.GetData("telefono_celular","SELECT telefono_celular FROM duenio WHERE dni_duenio = '"+dni+"'"));
    }
}

private void jcbPopliteos1ActionPerformed(java.awt.event.ActionEvent evt) {
    String popliteos = (String)jcbPopliteos1.getSelectedItem();
    String popliteos2 = "No";
    if(popliteos==popliteos2){
        jpPopliteos.setVisible(true);
    }else{
        jpPopliteos.setVisible(false);
    }
}

private void jcbInguinales1ActionPerformed(java.awt.event.ActionEvent evt) {
    String inguinales = (String)jcbInguinales1.getSelectedItem();
    String inguinales2 = "No";
    if(inguinales==inguinales2){
        jpInguinales.setVisible(true);
    }else{
        jpInguinales.setVisible(false);
    }
}

private void jcbAxilar1ActionPerformed(java.awt.event.ActionEvent evt) {
    String axilar = (String)jcbAxilar1.getSelectedItem();
    String axilar2 = "No";
    if(axilar==axilar2){
        jpAxilar.setVisible(true);
    }else{
        jpAxilar.setVisible(false);
    }
}

private void jcbSubmandibular1ActionPerformed(java.awt.event.ActionEvent evt) {
    String submandibular = (String)jcbSubmandibular1.getSelectedItem();

```



```
String submandibular2 = "No";
if(submandibular==submandibular2){
    jpSubmandibular.setVisible(true);
}else{
    jpSubmandibular.setVisible(false);
}
}
private void jcbPulsoRitmoActionPerformed(java.awt.event.ActionEvent evt) {
}
private void jtfFRActionPerformed(java.awt.event.ActionEvent evt) {
}
private void jcbHidratacionActionPerformed(java.awt.event.ActionEvent evt) {
    String hidratado="Deshidratado";
    String hidratado2=(String)jcbHidratacion.getSelectedItem();
    if (hidratado2==hidratado){
        jpDeshidratado.setVisible(true);
    }else{
        jpDeshidratado.setVisible(false);
    }
}
private void jbSalirActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}
MySQL animal = new MySQL();
// Variables declaration - do not modify
public javax.swing.JLabel jLabel1;
public javax.swing.JLabel jLabel10;
public javax.swing.JLabel jLabel11;
public javax.swing.JLabel jLabel116;
public javax.swing.JLabel jLabel117;
public javax.swing.JLabel jLabel118;
public javax.swing.JLabel jLabel119;
public javax.swing.JLabel jLabel12;
public javax.swing.JLabel jLabel120;
public javax.swing.JLabel jLabel121;
public javax.swing.JLabel jLabel122;
public javax.swing.JLabel jLabel123;
public javax.swing.JLabel jLabel124;
public javax.swing.JLabel jLabel125;
public javax.swing.JLabel jLabel13;
public javax.swing.JLabel jLabel14;
public javax.swing.JLabel jLabel143;
public javax.swing.JLabel jLabel15;
public static javax.swing.JLabel jLabel16;
public javax.swing.JLabel jLabel17;
public javax.swing.JLabel jLabel18;
public javax.swing.JLabel jLabel19;
public javax.swing.JLabel jLabel20;
public javax.swing.JLabel jLabel21;
public javax.swing.JLabel jLabel22;
public javax.swing.JLabel jLabel23;
public javax.swing.JLabel jLabel24;
public javax.swing.JLabel jLabel25;
public javax.swing.JLabel jLabel26;
public javax.swing.JLabel jLabel27;
public javax.swing.JLabel jLabel28;
public javax.swing.JLabel jLabel29;
public javax.swing.JLabel jLabel3;
public javax.swing.JLabel jLabel30;
public javax.swing.JLabel jLabel31;
public javax.swing.JLabel jLabel32;
public javax.swing.JLabel jLabel33;
```

```
public javax.swing.JLabel jLabel34;
public javax.swing.JLabel jLabel35;
public javax.swing.JLabel jLabel36;
public javax.swing.JLabel jLabel37;
public javax.swing.JLabel jLabel38;
public javax.swing.JLabel jLabel39;
public javax.swing.JLabel jLabel4;
public javax.swing.JLabel jLabel40;
public javax.swing.JLabel jLabel41;
public javax.swing.JLabel jLabel42;
public javax.swing.JLabel jLabel43;
public javax.swing.JLabel jLabel44;
public javax.swing.JLabel jLabel45;
public javax.swing.JLabel jLabel46;
public javax.swing.JLabel jLabel47;
public javax.swing.JLabel jLabel48;
public javax.swing.JLabel jLabel5;
public javax.swing.JLabel jLabel6;
public javax.swing.JLabel jLabel7;
public javax.swing.JLabel jLabel8;
public javax.swing.JLabel jLabel9;
public javax.swing.JScrollPane jScrollPane1;
public javax.swing.JScrollPane jScrollPane2;
public javax.swing.JScrollPane jScrollPane3;
public javax.swing.JScrollPane jScrollPane31;
public javax.swing.JScrollPane jScrollPane32;
public javax.swing.JScrollPane jScrollPane33;
public javax.swing.JScrollPane jScrollPane34;
public javax.swing.JScrollPane jScrollPane35;
public javax.swing.JScrollPane jScrollPane36;
public static javax.swing.JScrollPane jScrollPane37;
public static javax.swing.JTabbedPane jTabbedPane1;
public javax.swing.JButton jbGuardar;
public javax.swing.JButton jbSalir;
public javax.swing.JComboBox jcbAxilar1;
public javax.swing.JComboBox jcbAxilar2;
public javax.swing.JComboBox jcbAxilar3;
public javax.swing.JComboBox jcbAxilar4;
public javax.swing.JComboBox jcbCC;
public javax.swing.JComboBox jcbDatosDueño;
public javax.swing.JComboBox jcbHidratacion;
public javax.swing.JComboBox jcbInguinales1;
public javax.swing.JComboBox jcbInguinales2;
public javax.swing.JComboBox jcbInguinales3;
public javax.swing.JComboBox jcbInguinales4;
public javax.swing.JComboBox jcbLLCap;
public javax.swing.JComboBox jcbMaBucal;
public javax.swing.JComboBox jcbMaGenital;
public javax.swing.JComboBox jcbMaNasal;
public javax.swing.JComboBox jcbMaOcular;
public javax.swing.JComboBox jcbPopliteos1;
public javax.swing.JComboBox jcbPopliteos2;
public javax.swing.JComboBox jcbPopliteos3;
public javax.swing.JComboBox jcbPopliteos4;
public javax.swing.JComboBox jcbPulsoIntensidad;
public javax.swing.JComboBox jcbPulsoRitmo;
public javax.swing.JComboBox jcbSubmandibular1;
public javax.swing.JComboBox jcbSubmandibular2;
public javax.swing.JComboBox jcbSubmandibular3;
public javax.swing.JComboBox jcbSubmandibular4;
public static javax.swing.JLabel jlAlumno;
public static javax.swing.JLabel jlAlumno1;
```

```
public static javax.swing.JLabel jlAlumno2;
public static javax.swing.JLabel jlApellido;
public javax.swing.JLabel jlCasoClinico;
public static javax.swing.JLabel jlColor;
public javax.swing.JLabel jlDato;
public static javax.swing.JLabel jlDni;
public static javax.swing.JLabel jlDocente;
public static javax.swing.JLabel jlDocente1;
public static javax.swing.JLabel jlDocente2;
public static javax.swing.JLabel jlDomicilio;
public static javax.swing.JLabel jlEspecie;
public javax.swing.JLabel jlFecha;
public static javax.swing.JLabel jlFechaNacimiento;
public static javax.swing.JLabel jlNombre;
public static javax.swing.JLabel jlPeso;
public static javax.swing.JLabel jlRaza;
public static javax.swing.JLabel jlSexo;
public static javax.swing.JLabel jlTelefono;
public javax.swing.JPanel jpAxilar;
public javax.swing.JPanel jpConsulta1;
public javax.swing.JPanel jpConsulta10;
public javax.swing.JPanel jpConsulta2;
public javax.swing.JPanel jpConsulta5;
public javax.swing.JPanel jpDeshidratado;
public static javax.swing.JPanel jpGeneral;
public javax.swing.JPanel jpInguinales;
public javax.swing.JPanel jpPopliteos;
public javax.swing.JPanel jpSubmandibular;
public javax.swing.JTextArea jtaAnamnesis;
public javax.swing.JTextArea jtaCodExámenes;
public javax.swing.JTextArea jtaDiagnosticoDiferencial;
public javax.swing.JTextArea jtaDiagnosticoPresuntivo;
public javax.swing.JTextArea jtaDiagnosticoPronostico;
public javax.swing.JTextArea jtaHallazgos;
public javax.swing.JTextArea jtaIndicaciones;
public javax.swing.JTextArea jtaMotivo;
public javax.swing.JTextArea jtaTratamiento;
public javax.swing.JTextField jtfDeshidratado;
public javax.swing.JTextField jtfFC;
public javax.swing.JTextField jtfFR;
public javax.swing.JTextField jtfTemperatura;
public javax.swing.JTabbedPane jtpPestañas;
// End of variables declaration
}
```

MostrarConsultaAnimalP.Java

```
package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import java.beans.PropertyVetoException;
import java.sql.Date;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFrame;
import javax.swing.JInternalFrame;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
public class MostrarConsultaAnimalP extends javax.swing.JInternalFrame {
    public MostrarConsultaAnimalP() {
        initComponents();
        String cc = BuscarMostrarCasoClinico.auxenviar.getText();
        String fecha1 = BuscarMostrarCasoClinico.auxenviar1.getText();
        Date fecha = Date.valueOf(animal.GetData("fecha", "SELECT fecha FROM consulta WHERE caso_clinico = \""+cc+"\""));
        jlFecha.setText(fecha.toString());
        jlCasoClinico.setText(BuscarMostrarCasoClinico.auxenviar.getText()+"/"+fecha1);
        int cod_animal = Integer.parseInt(animal.GetData("cod_animal", "SELECT cod_animal FROM consulta WHERE caso_clinico = \""+cc+"\""));
        jlNombre.setText(animal.GetData("nombre", "SELECT nombre FROM animal_pequenos WHERE cod_animal = \""+cod_animal+"\""));
        int especie = Integer.parseInt(animal.GetData("cod_especie", "SELECT cod_especie FROM animal_pequenos WHERE cod_animal = \""+cod_animal+"\""));
        jlEspecie.setText(animal.GetData("especie", "SELECT especie FROM lista_especie WHERE cod_especie = \""+especie+"\""));
        int raza = Integer.parseInt(animal.GetData("cod_raza", "SELECT cod_raza FROM animal_pequenos WHERE cod_animal = \""+cod_animal+"\""));
        jlRaza.setText(animal.GetData("raza", "SELECT raza FROM lista_raza WHERE cod_raza = \""+raza+"\""));
        jlPeso.setText(animal.GetData("peso", "SELECT peso FROM animal_pequenos WHERE cod_animal = \""+cod_animal+"\""));
        jlFechaNacimiento.setText(animal.GetData("nacimiento", "SELECT nacimiento FROM animal_pequenos WHERE cod_animal = \""+cod_animal+"\""));
        jlSexo.setText(animal.GetData("sexo", "SELECT sexo FROM animal_pequenos WHERE cod_animal = \""+cod_animal+"\""));
        jlColor.setText(animal.GetData("color", "SELECT color FROM animal_pequenos WHERE cod_animal = \""+cod_animal+"\""));
        int duenio = Integer.parseInt(animal.GetData("dni_duenio", "SELECT dni_duenio FROM animal_pequenos WHERE cod_animal = \""+cod_animal+"\""));
        jlDni.setText(animal.GetData("dni_duenio", "SELECT dni_duenio FROM animal_pequenos WHERE cod_animal = \""+cod_animal+"\""));
        jlApellido.setText(animal.GetData("apellido", "SELECT apellido FROM duenio WHERE dni_duenio = \""+duenio+"\""));
        jlDomicilio.setText(animal.GetData("domicilio", "SELECT domicilio FROM duenio WHERE dni_duenio = \""+duenio+"\""));
        jlTelefono.setText(animal.GetData("telefono_fijo", "SELECT telefono_fijo FROM duenio WHERE dni_duenio = \""+duenio+"\""));
        jlDato.setText(animal.GetData("email", "SELECT email FROM duenio WHERE dni_duenio = \""+duenio+"\""));
        int dni_doctor = Integer.parseInt(animal.GetData("dni_doctor", "SELECT dni_doctor FROM consulta WHERE caso_clinico = \""+cc+"\""));
        jlDocente2.setText(String.valueOf(dni_doctor));
        jlDocente.setText(animal.GetData("apellido", "SELECT apellido FROM doctor WHERE dni_doctor = \""+dni_doctor+"\""));
        jlDocente1.setText(animal.GetData("nombre", "SELECT nombre FROM doctor WHERE dni_doctor = \""+dni_doctor+"\""));
        int dni_alumno = Integer.parseInt(animal.GetData("dni_alumno", "SELECT dni_alumno FROM consulta WHERE caso_clinico = \""+cc+"\""));
        jlAlumno2.setText(String.valueOf(dni_doctor));
        jlAlumno.setText(animal.GetData("apellido", "SELECT apellido FROM alumno WHERE dni_alumno = \""+dni_alumno+"\""));
        jlAlumno1.setText(animal.GetData("nombre", "SELECT nombre FROM alumno WHERE dni_alumno = \""+dni_alumno+"\""));
        jtaMotivo.setText(animal.GetData("motivo", "SELECT motivo FROM consulta WHERE caso_clinico = \""+cc+"\""));
        jtaAnamnesis.setText(animal.GetData("anamnesis", "SELECT anamnesis FROM consulta WHERE caso_clinico = \""+cc+"\""));
        jtfTemperatura.setText(animal.GetData("temperatura", "SELECT temperatura FROM consulta WHERE caso_clinico = \""+cc+"\""));
        jtfFC.setText(animal.GetData("fc", "SELECT fc FROM consulta WHERE caso_clinico = \""+cc+"\""));
        jtfFR.setText(animal.GetData("fr", "SELECT fr FROM consulta WHERE caso_clinico = \""+cc+"\""));
        jcbCC.setSelectedItem(animal.GetData("cc", "SELECT cc FROM consulta WHERE caso_clinico = \""+cc+"\""));
        jcbLLCap.setSelectedItem(animal.GetData("ll.cap", "SELECT 'll.cap' FROM consulta WHERE caso_clinico = \""+cc+"\""));
    }
}
```

```

jcbPulsoIntensidad.setSelectedItem(animal.GetData("pulso_intensidad","SELECT pulso_intensidad FROM consulta WHERE caso_clinico = \"+cc+\""));
jcbPulsoRitmo.setSelectedItem(animal.GetData("pulso_Ritmo","SELECT pulso_Ritmo FROM consulta WHERE caso_clinico = \"+cc+\""));
String hidratacion2="Normal";
String hidratacion=(animal.GetData("hidratacion","SELECT hidratacion FROM consulta WHERE caso_clinico = \"+cc+\""));
if (hidratacion2.equals(hidratacion)){
    jcbHidratacion.setSelectedItem(hidratacion);
}else{
    jtfDeshidratado.setText(hidratacion);
    jcbHidratacion.setSelectedItem("Deshidratado");
}
jcbMaOcular.setSelectedItem(animal.GetData("ma_ocular","SELECT ma_ocular FROM consulta WHERE caso_clinico = \"+cc+\""));
jcbMaBucal.setSelectedItem(animal.GetData("ma_bucal","SELECT ma_bucal FROM consulta WHERE caso_clinico = \"+cc+\""));
jcbMaNasal.setSelectedItem(animal.GetData("ma_nasal","SELECT ma_nasal FROM consulta WHERE caso_clinico = \"+cc+\""));
jcbMaGenital.setSelectedItem(animal.GetData("ma_genital","SELECT ma_genital FROM consulta WHERE caso_clinico = \"+cc+\""));
String submandibular = (animal.GetData("submandibular","SELECT submandibular FROM consulta WHERE caso_clinico = \"+cc+\""));
String submandibular2 = "Si";
String delimitadores= ",";
String[] palabrasSeparadas = submandibular.split(delimitadores);
if(submandibular2.equals(submandibular)){
}else{
    jcbSubmandibular1.setSelectedItem("No");
    jcbSubmandibular2.setSelectedItem(palabrasSeparadas[0]);
    jcbSubmandibular3.setSelectedItem(palabrasSeparadas[1]);
    jcbSubmandibular4.setSelectedItem(palabrasSeparadas[2]);
}
String axilar = (animal.GetData("axilar","SELECT axilar FROM consulta WHERE caso_clinico = \"+cc+\""));
String axilar2 = "Si";
if(axilar.equals(axilar2)){
}else{
    jcbAxilar1.setSelectedItem("No");
    palabrasSeparadas = axilar.split(delimitadores);
    jcbAxilar2.setSelectedItem(palabrasSeparadas[0]);
    jcbAxilar3.setSelectedItem(palabrasSeparadas[1]);
    jcbAxilar4.setSelectedItem(palabrasSeparadas[2]);
}
String inguinales = (animal.GetData("inguinales","SELECT inguinales FROM consulta WHERE caso_clinico = \"+cc+\""));
String inguinales2 = "Si";
if(inguinales.equals(inguinales2)){
}else{
    jcbSubmandibular1.setSelectedItem("No");
    palabrasSeparadas = inguinales.split(delimitadores);
    jcbInguinales2.setSelectedItem(palabrasSeparadas[0]);
    jcbInguinales3.setSelectedItem(palabrasSeparadas[1]);
    jcbInguinales4.setSelectedItem(palabrasSeparadas[2]);
}
String popliteos = (animal.GetData("popliteos","SELECT popliteos FROM consulta WHERE caso_clinico = \"+cc+\""));
String popliteos2 = "Si";
if(popliteos.equals(popliteos2)){
}else{
    jcbPopliteos1.setSelectedItem("No");
    palabrasSeparadas = popliteos.split(delimitadores);
    jcbPopliteos2.setSelectedItem(palabrasSeparadas[0]);
    jcbPopliteos3.setSelectedItem(palabrasSeparadas[1]);
    jcbPopliteos4.setSelectedItem(palabrasSeparadas[2]);
}
String lesion = (animal.GetData("lesion_simetria","SELECT lesion_simetria FROM consulta WHERE caso_clinico = \"+cc+\""));
String lesion2 = "";
if(lesion.equals(lesion2)){jcbLesion.setSelectedItem("No");}else{
    jcbLesion.setSelectedItem("Si");
    jtaLesionTipo.setText(animal.GetData("lesion_tipo","SELECT lesion_tipo FROM consulta WHERE caso_clinico = \"+cc+\""));
    jtfLesionForma.setText(animal.GetData("lesion_forma","SELECT lesion_forma FROM consulta WHERE caso_clinico = \"+cc+\""));
}

```



```

jcbLesionSimetria.setSelectedItem(animal.GetData("lesion_simetria","SELECT lesion_simetria FROM consulta WHERE caso_clinico = \"+cc+\""));
jtaLesionOlor.setText(animal.GetData("olor","SELECT olor FROM consulta WHERE caso_clinico = \"+cc+\""));
jcbLesionPrurito.setSelectedItem(animal.GetData("prurito","SELECT prurito FROM consulta WHERE caso_clinico = \"+cc+\""));
jtaLesionManto.setText(animal.GetData("manto_piloso","SELECT manto_piloso FROM consulta WHERE caso_clinico = \"+cc+\""));
jcbLesionEctoparasitos.setSelectedItem(animal.GetData("ectoparasitos","SELECT ectoparasitos FROM consulta WHERE caso_clinico = \"+cc+\""));
jtaLesionOidos.setText(animal.GetData("oido","SELECT oido FROM consulta WHERE caso_clinico = \"+cc+\""));
}
jtaBoca.setText(animal.GetData("boca","SELECT boca FROM consulta WHERE caso_clinico = \"+cc+\""));
jtaEsófago.setText(animal.GetData("esofago","SELECT esofago FROM consulta WHERE caso_clinico = \"+cc+\""));
jtaEstomago.setText(animal.GetData("estomago","SELECT estomago FROM consulta WHERE caso_clinico = \"+cc+\""));
jtaIntestinos.setText(animal.GetData("intestino","SELECT intestino FROM consulta WHERE caso_clinico = \"+cc+\""));
jtaHigado.setText(animal.GetData("higado","SELECT higado FROM consulta WHERE caso_clinico = \"+cc+\""));
jcbRegurgitacion.setSelectedItem(animal.GetData("regurgitacion","SELECT regurgitacion FROM consulta WHERE caso_clinico = \"+cc+\""));
jcbVomito.setSelectedItem(animal.GetData("vomito","SELECT vomito FROM consulta WHERE caso_clinico = \"+cc+\""));
jcbDiarrea.setSelectedItem(animal.GetData("diarrea","SELECT diarrea FROM consulta WHERE caso_clinico = \"+cc+\""));
jcbRuidos.setSelectedItem(animal.GetData("ruidos","SELECT ruidos FROM consulta WHERE caso_clinico = \"+cc+\""));
jcbDistension.setSelectedItem(animal.GetData("distension","SELECT distension FROM consulta WHERE caso_clinico = \"+cc+\""));
jtaViasSuperiores.setText(animal.GetData("vias_superiores","SELECT vias_superiores FROM consulta WHERE caso_clinico = \"+cc+\""));
jtaViasInferiores.setText(animal.GetData("vias_inferiores","SELECT vias_inferiores FROM consulta WHERE caso_clinico = \"+cc+\""));
jcbRitmoRespiratorio.setSelectedItem(animal.GetData("ritmo_respiratorio","SELECT ritmo_respiratorio FROM consulta WHERE caso_clinico = \"+cc+\""));
jcbTipo.setSelectedItem(animal.GetData("tipo","SELECT tipo FROM consulta WHERE caso_clinico = \"+cc+\""));
jcbTosReflejo.setSelectedItem(animal.GetData("tos_reflejo","SELECT tos_reflejo FROM consulta WHERE caso_clinico = \"+cc+\""));
jtaAuscultacion.setText(animal.GetData("auscultacion","SELECT auscultacion FROM consulta WHERE caso_clinico = \"+cc+\""));
jtaRiñones.setText(animal.GetData("riñones","SELECT riñones FROM consulta WHERE caso_clinico = \"+cc+\""));
jtaUreteres.setText(animal.GetData("ureteres","SELECT ureteres FROM consulta WHERE caso_clinico = \"+cc+\""));
jtaVejiga.setText(animal.GetData("vejiga","SELECT vejiga FROM consulta WHERE caso_clinico = \"+cc+\""));
jtaUretra.setText(animal.GetData("uretra","SELECT uretra FROM consulta WHERE caso_clinico = \"+cc+\""));
String secreciones=(animal.GetData("secreciones","SELECT secreciones FROM consulta WHERE caso_clinico = \"+cc+\""));
String secreciones2 = "";
if(secreciones.equals(secreciones2)){
    jpSecreciones.setVisible(false);
    jcbSecreciones.setSelectedItem("No");
}else{
    jpSecreciones.setVisible(true);
    jcbSecreciones.setSelectedItem("Si");
    jtaSecreciones.setText(secreciones);
}
jtaOrina.setText(animal.GetData("orina","SELECT orina FROM consulta WHERE caso_clinico = \"+cc+\""));
jtaCorazonAuscultacion.setText(animal.GetData("corazon_auscultacion","SELECT corazon_auscultacion FROM consulta WHERE caso_clinico = \"+cc+\""));
jcbRitmoCorazon.setSelectedItem(animal.GetData("ritmo_corazon","SELECT ritmo_corazon FROM consulta WHERE caso_clinico = \"+cc+\""));
jtaGenitalesInterno.setText(animal.GetData("genitales_interno","SELECT genitales_interno FROM consulta WHERE caso_clinico = \"+cc+\""));
jtaGenitalesExterno.setText(animal.GetData("genitales_externo","SELECT genitales_externo FROM consulta WHERE caso_clinico = \"+cc+\""));
String genitales_secreciones = (animal.GetData("genitales_secreciones","SELECT genitales_secreciones FROM consulta WHERE caso_clinico = \"+cc+\""));
String genitales_secreciones2 = "No";
if(genitales_secreciones.equals(genitales_secreciones2)){
    jcbGenitalesSecreciones.setSelectedItem("No");
}else{
    jtaGenitalesSecreciones.setText(animal.GetData("genitales_secreciones","SELECT genitales_secreciones FROM consulta WHERE caso_clinico = \"+cc+\""));
    jcbGenitalesSecreciones.setSelectedItem("Si");
}
jtaTactoRectal.setText(animal.GetData("tacto_rectal","SELECT tacto_rectal FROM consulta WHERE caso_clinico = \"+cc+\""));
jtaLocomotorLesion.setText(animal.GetData("locomotor_lesion","SELECT locomotor_lesion FROM consulta WHERE caso_clinico = \"+cc+\""));
jtaLocomotorUbicacionL.setText(animal.GetData("locomotor_ubicacionL","SELECT locomotor_ubicacionL FROM consulta WHERE caso_clinico = \"+cc+\""));
jtaLocomotorDesformacion.setText(animal.GetData("locomotor_deformacion","SELECT locomotor_deformacion FROM consulta WHERE caso_clinico = \"+cc+\""));

```

```

jtaLocomotorUbicacionD.setText(animal.GetData("locomotor_ubicacionD","SELECT locomotor_ubicacionD FROM consulta WHERE caso_clinico = \"+cc+\""));
String claudicacion_tipo = (animal.GetData("claudicacion_tipo","SELECT claudicacion_tipo FROM consulta WHERE caso_clinico = \"+cc+\""));
String claudicacion_tipo2 = "0";
if(claudicacion_tipo.equals(claudicacion_tipo2)){
    jcbClaudicacion1.setSelectedItem("No");
    jcbClaudicacion2.setSelectedItem(claudicacion_tipo);
}else{
    jcbClaudicacion1.setSelectedItem("Si");
    jcbClaudicacion2.setSelectedItem(claudicacion_tipo);
}
jcbClaudicacionMiembro.setSelectedItem(animal.GetData("claudicacion_miembro","SELECT claudicacion_miembro FROM consulta WHERE caso_clinico = \"+cc+\""));
jcbNerviosoUbicacion.setSelectedItem(animal.GetData("nervioso_ubicacion","SELECT nervioso_ubicacion FROM consulta WHERE caso_clinico = \"+cc+\""));
String paralisis = (animal.GetData("paralisis","SELECT paralisis FROM consulta WHERE caso_clinico = \"+cc+\""));
String paralisis2 = "No";
if(paralisis.equals(paralisis2)){
    jcbParalisis.setSelectedItem("NO");
}else{
    jtfParalisis.setText(paralisis);
    jcbParalisis.setSelectedItem("Si");
}
String paresia = (animal.GetData("paresia","SELECT paresia FROM consulta WHERE caso_clinico = \"+cc+\""));
String paresia2 = "No";
if(paresia == paresia2){
    jcbParesia.setSelectedItem("No");
}else{
    jtfParesia.setText(paresia);
    jcbParesia.setSelectedItem("Si");
}
String convulsion = (animal.GetData("convulsion","SELECT convulsion FROM consulta WHERE caso_clinico = \"+cc+\""));
String convulsion2 = "No";
if(convulsion == convulsion2){
    jcbConvulsion.setSelectedItem("No");
}else{
    jcbConvulsion.setSelectedItem("Si");
    jtfConvulsion.setText(convulsion);
}
jcbAtaxia.setSelectedItem(animal.GetData("ataxia","SELECT ataxia FROM consulta WHERE caso_clinico = \"+cc+\""));
jtfReflejos.setText(animal.GetData("reflejos","SELECT reflejos FROM consulta WHERE caso_clinico = \"+cc+\""));
jcbSensibilidad.setSelectedItem(animal.GetData("sensibilidad","SELECT sensibilidad FROM consulta WHERE caso_clinico = \"+cc+\""));
jcbConducta.setSelectedItem(animal.GetData("conducta","SELECT conducta FROM consulta WHERE caso_clinico = \"+cc+\""));
jcbEstadoSensorio.setSelectedItem(animal.GetData("estado_sensorio","SELECT estado_sensorio FROM consulta WHERE caso_clinico = \"+cc+\""));
jcbSNMS.setSelectedItem(animal.GetData("snms","SELECT snms FROM consulta WHERE caso_clinico = \"+cc+\""));
jcbSNMI.setSelectedItem(animal.GetData("snmi","SELECT snmi FROM consulta WHERE caso_clinico = \"+cc+\""));
jtfOjoDerecho.setText(animal.GetData("ojo_derecho","SELECT ojo_derecho FROM consulta WHERE caso_clinico = \"+cc+\""));
jtfOjoIzquierdo.setText(animal.GetData("ojo_izq","SELECT ojo_izq FROM consulta WHERE caso_clinico = \"+cc+\""));
jtaDiagnosticoPresuntivo.setText(animal.GetData("diagnostico_presuntivo","SELECT diagnostico_presuntivo FROM consulta WHERE caso_clinico = \"+cc+\""));
jtaDiagnosticoDiferencial.setText(animal.GetData("diagnostico_diferencial","SELECT diagnostico_diferencial FROM consulta WHERE caso_clinico = \"+cc+\""));
jtaDiagnosticoPronostico.setText(animal.GetData("diagnostico_pronostico","SELECT diagnostico_pronostico FROM consulta WHERE caso_clinico = \"+cc+\""));
jtaTratamiento.setText(animal.GetData("tratamiento","SELECT tratamiento FROM consulta WHERE caso_clinico = \"+cc+\""));
jtaIndicaciones.setText(animal.GetData("indicaciones","SELECT indicaciones FROM consulta WHERE caso_clinico = \"+cc+\""));
String lesion_ubicacion="dds";
}
/*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")
private void jcbGenitalesSecrecionesActionPerformed(java.awt.event.ActionEvent evt) {
    String genitalesSecreciones = (String)jcbGenitalesSecreciones.getSelectedItemAt();
    String genitalesSecreciones2 = "Si";
    if(genitalesSecreciones==genitalesSecreciones2){
        jpGenitalesSecreciones.setVisible(true);
    }else{
        jpGenitalesSecreciones.setVisible(false);
    }
}

private void jtfLesionFormaActionPerformed(java.awt.event.ActionEvent evt) {
}

private void jcbHidratacionActionPerformed(java.awt.event.ActionEvent evt) {
    String hidratado="Deshidratado";
    String hidratado2=(String)jcbHidratacion.getSelectedItemAt();
    if (hidratado2==hidratado){
        jpDeshidratado.setVisible(true);
    }else{
        jpDeshidratado.setVisible(false);
    }
}

private void jcbSubmandibular1ActionPerformed(java.awt.event.ActionEvent evt) {
    String submandibular = (String)jcbSubmandibular1.getSelectedItemAt();
    String submandibular2 = "No";
    if(submandibular==submandibular2){
        jpSubmandibular.setVisible(true);
    }else{
        jpSubmandibular.setVisible(false);
    }
}

private void jcbAxilar1ActionPerformed(java.awt.event.ActionEvent evt) {
    String axilar = (String)jcbAxilar1.getSelectedItemAt();
    String axilar2 = "No";
    if(axilar==axilar2){
        jpAxilar.setVisible(true);
    }else{
        jpAxilar.setVisible(false);
    }
}

private void jcbInguinales1ActionPerformed(java.awt.event.ActionEvent evt) {
    String inguinales = (String)jcbInguinales1.getSelectedItemAt();
    String inguinales2 = "No";
    if(inguinales==inguinales2){
        jpInguinales.setVisible(true);
    }else{
        jpInguinales.setVisible(false);
    }
}

private void jcbPopliteos1ActionPerformed(java.awt.event.ActionEvent evt) {
    String popliteos = (String)jcbPopliteos1.getSelectedItemAt();
    String popliteos2 = "No";
    if(popliteos==popliteos2){
        jpPopliteos.setVisible(true);
    }else{
        jpPopliteos.setVisible(false);
    }
}

private void jcbSecrecionesActionPerformed(java.awt.event.ActionEvent evt) {
    String secreciones = (String)jcbSecreciones.getSelectedItemAt();
    String secreciones2 = "Si";
    if(secreciones==secreciones2){
        jpSecreciones.setVisible(true);
    }
}
```

```
}else{
    jpSecreciones.setVisible(false);
}
}

private void jcbLesionActionPerformed(java.awt.event.ActionEvent evt) {
    String lesion = (String)jcbLesion.getSelectedItem();
    String lesion2 = "Si";
    if(lesion==lesion2){
        jpLesion.setVisible(true);
    }else{
        jpLesion.setVisible(false);
    }
}

private void jcbClaudicacion1ActionPerformed(java.awt.event.ActionEvent evt) {
    String claudicacion = (String)jcbClaudicacion1.getSelectedItem();
    String claudicacion2 = "Si";
    if(claudicacion==claudicacion2){
        jpClaudicacion.setVisible(true);
    }else{
        jpClaudicacion.setVisible(false);
    }
}

private void jcbParalisisActionPerformed(java.awt.event.ActionEvent evt) {
    String paralisis = (String)jcbParalisis.getSelectedItem();
    String paralisis2 = "Si";
    if(paralisis==paralisis2){
        jpParalisis.setVisible(true);
    }else{
        jpParalisis.setVisible(false);
    }
}

private void jcbParesiaActionPerformed(java.awt.event.ActionEvent evt) {
    String paresia = (String)jcbParesia.getSelectedItem();
    String paresia2 = "Si";
    if(paresia ==paresia2){
        jpParesia.setVisible(true);
    }else{
        jpParesia.setVisible(false);
    }
}

private void jcbConvulsionActionPerformed(java.awt.event.ActionEvent evt) {
    String convulsion = (String)jcbConvulsion.getSelectedItem();
    String convulsion2 = "Si";
    if(convulsion ==convulsion2){
        jpConvulsion.setVisible(true);
    }else{
        jpConvulsion.setVisible(false);
    }
}

private void jbGuardarActionPerformed(java.awt.event.ActionEvent evt) {
    String caso_clinico = jlcasoClinico.getText();
    String cod_animal="1";
    String dni_doctor=jldoctor2.getText();
    String dni_alumno=jlalumno2.getText();
    String lesion_tipo="";
    String lesion_forma="";
    String lesion_simetria="";
    String olor="";
    String prurito="";
    String manto_piloso="";
    String ectoparasitos="";
    String oido="";
```

```
boolean aux = animal.existe("consulta", "caso_clinico", caso_clinico);
if (aux==true){
    JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
}else{
    String motivo = jtaMotivo.getText();
    String anamnesis = jtaAnamnesis.getText();
    String temperatura = jtfTemperatura.getText();
    String fc = jtfFC.getText();
    String fr = jtfFR.getText();
    String cc = (String)jcbCC.getSelectedItemAt();
    String ll_cap = (String)jcbLLCap.getSelectedItemAt();
    String pulso_ritmo = (String)jcbPulsoRitmo.getSelectedItemAt();
    String pulso_intensidad = (String)jcbPulsoIntensidad.getSelectedItemAt();
    String hidratacion2="Deshidratado";
    String hidratacion=(String)jcbHidratacion.getSelectedItemAt();
    if (hidratacion2==hidratacion){
        hidratacion = jtfDeshidratado.getText();
    }else{
        hidratacion=(String)jcbHidratacion.getSelectedItemAt();
    }
    String ma_ocular = (String)jcbMaOcular.getSelectedItemAt();
    String ma_bucal = (String)jcbMaBucal.getSelectedItemAt();
    String ma_nasal = (String)jcbMaNasal.getSelectedItemAt();
    String ma_genital = (String)jcbMaGenital.getSelectedItemAt();
    String submandibular = (String)jcbSubmandibular1.getSelectedItemAt();
    String submandibular2 = "No";
    if(submandibular==submandibular2){
        submandibular =
        (String)jcbSubmandibular2.getSelectedItemAt()+" "+jcbSubmandibular3.getSelectedItemAt()+" "+jcbSubmandibular4.getSelectedItemAt();
    }else{
        submandibular = "Si";
    }
    String axilar = (String)jcbAxilar1.getSelectedItemAt();
    String axilar2 = "No";
    if(axilar==axilar2){
        axilar = (String)jcbAxilar2.getSelectedItemAt()+" "+jcbAxilar3.getSelectedItemAt()+" "+jcbAxilar4.getSelectedItemAt();
    }else{
        axilar = "Si";
    }
    String inguinales = (String)jcbInguinales1.getSelectedItemAt();
    String inguinales2 = "No";
    if(inguinales==inguinales2){
        inguinales = (String)jcbInguinales2.getSelectedItemAt()+" "+jcbInguinales3.getSelectedItemAt()+" "+jcbInguinales4.getSelectedItemAt();
    }else{
        inguinales = "Si";
    }
    String popliteos = (String)jcbPopliteos1.getSelectedItemAt();
    String popliteos2 = "No";
    if(popliteos==popliteos2){
        popliteos = (String)jcbPopliteos2.getSelectedItemAt()+" "+jcbPopliteos3.getSelectedItemAt()+" "+jcbPopliteos4.getSelectedItemAt();
    }else{
        popliteos = "Si";
    }
    String lesion = (String)jcbLesion.getSelectedItemAt();
    String lesion2 = "Si";
    if(lesion==lesion2){
        lesion_tipo = jtaLesionTipo.getText();
        lesion_forma = jtfLesionForma.getText();
        lesion_simetria = (String)jcbLesionSimetria.getSelectedItemAt();
        olor = jtaLesionOlor.getText();
        prurito = (String)jcbLesionPrurito.getSelectedItemAt();
        manto_piloso = jtaLesionManto.getText();
    }
```

```

    ectoparasitos = (String)jcbLesionEctoparasitos.getSelectedItemId();
    oido = jtaLesionOidos.getText();
}
String boca = jtaBoca.getText();
String esofago = jtaEsofago.getText();
String estomago = jtaEstomago.getText();
String intestino = jtaIntestinos.getText();
String higado = jtaHigado.getText();
String regurgitacion = (String)jcbRegurgitacion.getSelectedItemId();
String vomito = (String)jcbVomito.getSelectedItemId();
String diarrea = (String)jcbDiarrea.getSelectedItemId();
String ruidos = (String)jcbRuidos.getSelectedItemId();
String distension = (String)jcbDistension.getSelectedItemId();
String vias_superiores = jtaViasSuperiores.getText();
String vias_inferiores = jtaViasInferiores.getText();
String ritmo_respiratorio = (String)jcbRitmoRespiratorio.getSelectedItemId();
String tipo = (String)jcbTipo.getSelectedItemId();
String tos_reflejo = (String)jcbTosReflejo.getSelectedItemId();
String auscultacion = jtaAuscultacion.getText();
String riñones = jtaRiñones.getText();
String ureteres = jtaUreteres.getText();
String vejiga = jtaVejiga.getText();
String uretra = jtaUretra.getText();
String secreciones = jtaSecreciones.getText();
String orina = jtaOrina.getText();
String corazon_auscultacion = jtaCorazonAuscultacion.getText();
String ritmo_corazon = (String)jcbRitmoCorazon.getSelectedItemId();
String genitales_interno = jtaGenitalesInterno.getText();
String genitales_externo = jtaGenitalesExterno.getText();
String genitales_secreciones = (String)jcbGenitalesSecreciones.getSelectedItemId();
String genitales_secreciones2 = "Si";
if(genitales_secreciones==genitales_secreciones2){
    genitales_secreciones=jtaGenitalesSecreciones.getText();
}
String tacto_rectal = jtaTactoRectal.getText();
String locomotor_lesion = jtaLocomotorLesion.getText();
String locomotor_ubicacionL = jtaLocomotorUbicacionL.getText();
String locomotor_deformacion = jtaLocomotorDesformacion.getText();
String locomotor_ubicacionD = jtaLocomotorUbicacionD.getText();
String claudicacion_tipo = (String)jcbClaudicacion1.getSelectedItemId();
String claudicacion_tipo2 = "Si";
if(claudicacion_tipo==claudicacion_tipo2){
    claudicacion_tipo = (String)jcbClaudicacion2.getSelectedItemId();
}else{
    claudicacion_tipo = "0";
}
String claudicacion_miembro = (String)jcbClaudicacionMiembro.getSelectedItemId();
String nervioso_ubicacion = (String)jcbNerviosoUbicacion.getSelectedItemId();
String paralisis = (String)jcbParalisis.getSelectedItemId();
String paralisis2 = "Si";
if(paralisis==paralisis2){
    paralisis = jtfParalisis.getText();
}
String paresia = (String)jcbParesia.getSelectedItemId();
String paresia2 = "Si";
if(paresia ==paresia2){
    paresia = jtfParesia.getText();
}
String convulsion = (String)jcbConvulsion.getSelectedItemId();
String convulsion2 = "Si";
if(convulsion ==convulsion2){
    convulsion = jtfConvulsion.getText();
}

```

```

    }
    String ataxia = (String)jcbAtaxia.getSelectedItem();
    String reflejos = jtfReflejos.getText();
    String sensibilidad = (String)jcbSensibilidad.getSelectedItem();
    String conducta = (String)jcbConducta.getSelectedItem();
    String estado_sensorio = (String)jcbEstadoSensorio.getSelectedItem();
    String snms = (String)jcbSNMS.getSelectedItem();
    String snmi = (String)jcbSNMI.getSelectedItem();
    String ojo_derecho = jtfOjoDerecho.getText();
    String ojo_izq = jtfOjoIzquierdo.getText();
    String diagnostico_presuntivo = jtaDiagnosticoPresuntivo.getText();
    String diagnostico_diferencial = jtaDiagnosticoDiferencial.getText();
    String diagnostico_pronostico = jtaDiagnosticoPronostico.getText();
    String tratamiento = jtaTratamiento.getText();
    String indicaciones = jtaIndicaciones.getText();
    String lesion_ubicacion="dds";
    java.util.Date date = new java.util.Date();
    java.text.SimpleDateFormat sdf=new java.text.SimpleDateFormat("yyyy-MM-dd");
    String fecha = sdf.format(date);
    String datos[] =
    {caso_clinico,fecha,cod_animal,dni_doctor,dni_alumno,motivo,anamnesis,temperatura,fc,fr,cc,Il_cap,pulso_ritmo,pulso_intensidad,hidratacion,ma_
    ocular,ma_bucal,ma_nasal,ma_genital,submandibular,axilar,inguinales,popliteos,lesion_tipo,lesion_forma,lesion_ubicacion,lesion_simetria,olor,pruri
    to,manto_piloso,ectoparasitos,oido,boca,esofago,estomago,intestino,higado,regurgitacion,vomito,diarrea,ruidos,distension,vias_superiores,vias_infe
    riores,ritmo_respiratorio,tipo,tos_reflejo,auscultacion,corazon_auscultacion,ritmo_corazon,riñones,ureteres,vejiga,uretra,secreciones,orina,genitales
    _interno,genitales_externo,genitales_secreciones,tacto_rectal,locomotor_lesion,locomotor_ubicacionL,locomotor_deformacion,locomotor_ubicacio
    nD,claudicacion_miembro,claudicacion_tipo,nervioso_ubicacion,paralisis,paresia,convulsion,ataxia,reflejos,sensibilidad,conducta,estado_sensorio,sn
    ms,snmi,ojo_derecho,ojo_izq,diagnostico_presuntivo,diagnostico_diferencial,diagnostico_pronostico,tratamiento,indicaciones};
    animal.InsertarRegistro(datos, "insert into consulta
    values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,
    ?,?,?)");
    }
}

private void jcbDatosDueñoActionPerformed(java.awt.event.ActionEvent evt) {
    String dni= jIDni.getText();
    String dato = jcbDatosDueño.getSelectedItem().toString();
    if(dato=="e-mail"){
        jIDato.setText(animal.GetData("email","SELECT email FROM duenio WHERE dni_duenio = '"+dni+"'"));
    }
    if(dato=="facebook"){
        jIDato.setText(animal.GetData("facebook","SELECT facebook FROM duenio WHERE dni_duenio = '"+dni+"'"));
    }
    if(dato=="celular"){
        jIDato.setText(animal.GetData("telefono_celular","SELECT telefono_celular FROM duenio WHERE dni_duenio = '"+dni+"'"));
    }
}

private void jbSalirActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}

private void jcbPulsoIntensidadActionPerformed(java.awt.event.ActionEvent evt) {
}

MySQL animal = new MySQL();
// Variables declaration - do not modify
public javax.swing.JLabel jLabel1;
public javax.swing.JLabel jLabel10;
public javax.swing.JLabel jLabel100;
public javax.swing.JLabel jLabel101;
public javax.swing.JLabel jLabel102;
public javax.swing.JLabel jLabel103;
public javax.swing.JLabel jLabel104;
public javax.swing.JLabel jLabel105;
public javax.swing.JLabel jLabel106;
public javax.swing.JLabel jLabel107;

```

```
public javax.swing.JLabel jLabel108;
public javax.swing.JLabel jLabel109;
public javax.swing.JLabel jLabel11;
public javax.swing.JLabel jLabel110;
public javax.swing.JLabel jLabel111;
public javax.swing.JLabel jLabel112;
public javax.swing.JLabel jLabel113;
public javax.swing.JLabel jLabel114;
public javax.swing.JLabel jLabel115;
public javax.swing.JLabel jLabel116;
public javax.swing.JLabel jLabel117;
public javax.swing.JLabel jLabel118;
public javax.swing.JLabel jLabel119;
public javax.swing.JLabel jLabel12;
public javax.swing.JLabel jLabel120;
public javax.swing.JLabel jLabel121;
public javax.swing.JLabel jLabel122;
public javax.swing.JLabel jLabel123;
public javax.swing.JLabel jLabel124;
public javax.swing.JLabel jLabel125;
public javax.swing.JLabel jLabel13;
public javax.swing.JLabel jLabel14;
public javax.swing.JLabel jLabel143;
public javax.swing.JLabel jLabel15;
public static javax.swing.JLabel jLabel16;
public javax.swing.JLabel jLabel17;
public javax.swing.JLabel jLabel18;
public javax.swing.JLabel jLabel19;
public javax.swing.JLabel jLabel20;
public javax.swing.JLabel jLabel21;
public javax.swing.JLabel jLabel22;
public javax.swing.JLabel jLabel23;
public javax.swing.JLabel jLabel24;
public javax.swing.JLabel jLabel25;
public javax.swing.JLabel jLabel26;
public javax.swing.JLabel jLabel27;
public javax.swing.JLabel jLabel28;
public javax.swing.JLabel jLabel29;
public javax.swing.JLabel jLabel3;
public javax.swing.JLabel jLabel30;
public javax.swing.JLabel jLabel31;
public javax.swing.JLabel jLabel32;
public javax.swing.JLabel jLabel33;
public javax.swing.JLabel jLabel34;
public javax.swing.JLabel jLabel35;
public javax.swing.JLabel jLabel36;
public javax.swing.JLabel jLabel37;
public javax.swing.JLabel jLabel38;
public javax.swing.JLabel jLabel39;
public javax.swing.JLabel jLabel4;
public javax.swing.JLabel jLabel40;
public javax.swing.JLabel jLabel41;
public javax.swing.JLabel jLabel42;
public javax.swing.JLabel jLabel43;
public javax.swing.JLabel jLabel44;
public javax.swing.JLabel jLabel45;
public javax.swing.JLabel jLabel46;
public javax.swing.JLabel jLabel47;
public javax.swing.JLabel jLabel48;
public javax.swing.JLabel jLabel49;
public javax.swing.JLabel jLabel5;
public javax.swing.JLabel jLabel50;
```

```
public javax.swing.JLabel jLabel51;
public javax.swing.JLabel jLabel52;
public javax.swing.JLabel jLabel53;
public javax.swing.JLabel jLabel54;
public javax.swing.JLabel jLabel55;
public javax.swing.JLabel jLabel56;
public javax.swing.JLabel jLabel57;
public javax.swing.JLabel jLabel58;
public javax.swing.JLabel jLabel59;
public javax.swing.JLabel jLabel6;
public javax.swing.JLabel jLabel60;
public javax.swing.JLabel jLabel61;
public javax.swing.JLabel jLabel62;
public javax.swing.JLabel jLabel63;
public javax.swing.JLabel jLabel64;
public javax.swing.JLabel jLabel65;
public javax.swing.JLabel jLabel66;
public javax.swing.JLabel jLabel67;
public javax.swing.JLabel jLabel68;
public javax.swing.JLabel jLabel69;
public javax.swing.JLabel jLabel7;
public javax.swing.JLabel jLabel70;
public javax.swing.JLabel jLabel71;
public javax.swing.JLabel jLabel72;
public javax.swing.JLabel jLabel73;
public javax.swing.JLabel jLabel74;
public javax.swing.JLabel jLabel75;
public javax.swing.JLabel jLabel76;
public javax.swing.JLabel jLabel77;
public javax.swing.JLabel jLabel78;
public javax.swing.JLabel jLabel79;
public javax.swing.JLabel jLabel8;
public javax.swing.JLabel jLabel80;
public javax.swing.JLabel jLabel81;
public javax.swing.JLabel jLabel82;
public javax.swing.JLabel jLabel83;
public javax.swing.JLabel jLabel84;
public javax.swing.JLabel jLabel85;
public javax.swing.JLabel jLabel86;
public javax.swing.JLabel jLabel87;
public javax.swing.JLabel jLabel88;
public javax.swing.JLabel jLabel89;
public javax.swing.JLabel jLabel9;
public javax.swing.JLabel jLabel90;
public javax.swing.JLabel jLabel91;
public javax.swing.JLabel jLabel92;
public javax.swing.JLabel jLabel93;
public javax.swing.JLabel jLabel94;
public javax.swing.JLabel jLabel95;
public javax.swing.JLabel jLabel96;
public javax.swing.JLabel jLabel97;
public javax.swing.JLabel jLabel98;
public javax.swing.JLabel jLabel99;
public javax.swing.JScrollPane jScrollPane1;
public javax.swing.JScrollPane jScrollPane10;
public javax.swing.JScrollPane jScrollPane11;
public javax.swing.JScrollPane jScrollPane12;
public javax.swing.JScrollPane jScrollPane13;
public javax.swing.JScrollPane jScrollPane14;
public javax.swing.JScrollPane jScrollPane15;
public javax.swing.JScrollPane jScrollPane16;
public javax.swing.JScrollPane jScrollPane17;
```



```
public javax.swing.JScrollPane jScrollPane18;
public javax.swing.JScrollPane jScrollPane19;
public javax.swing.JScrollPane jScrollPane2;
public javax.swing.JScrollPane jScrollPane20;
public javax.swing.JScrollPane jScrollPane21;
public javax.swing.JScrollPane jScrollPane22;
public javax.swing.JScrollPane jScrollPane23;
public javax.swing.JScrollPane jScrollPane24;
public javax.swing.JScrollPane jScrollPane25;
public javax.swing.JScrollPane jScrollPane26;
public javax.swing.JScrollPane jScrollPane27;
public javax.swing.JScrollPane jScrollPane28;
public javax.swing.JScrollPane jScrollPane29;
public javax.swing.JScrollPane jScrollPane3;
public javax.swing.JScrollPane jScrollPane31;
public javax.swing.JScrollPane jScrollPane32;
public javax.swing.JScrollPane jScrollPane33;
public javax.swing.JScrollPane jScrollPane34;
public javax.swing.JScrollPane jScrollPane35;
public javax.swing.JScrollPane jScrollPane36;
public static javax.swing.JScrollPane jScrollPane37;
public javax.swing.JScrollPane jScrollPane4;
public javax.swing.JScrollPane jScrollPane5;
public javax.swing.JScrollPane jScrollPane6;
public javax.swing.JScrollPane jScrollPane7;
public javax.swing.JScrollPane jScrollPane8;
public javax.swing.JScrollPane jScrollPane9;
public static javax.swing.JTabbedPane jTabbedPane1;
public javax.swing.JButton jbGuardar;
public javax.swing.JButton jbSalir;
public javax.swing.JComboBox jcbAtaxia;
public javax.swing.JComboBox jcbAxilar1;
public javax.swing.JComboBox jcbAxilar2;
public javax.swing.JComboBox jcbAxilar3;
public javax.swing.JComboBox jcbAxilar4;
public javax.swing.JComboBox jcbCC;
public javax.swing.JComboBox jcbClaudicacion1;
public javax.swing.JComboBox jcbClaudicacion2;
public javax.swing.JComboBox jcbClaudicacionMiembro;
public javax.swing.JComboBox jcbConducta;
public javax.swing.JComboBox jcbConvulsion;
public javax.swing.JComboBox jcbDatosDueño;
public javax.swing.JComboBox jcbDiarrea;
public javax.swing.JComboBox jcbDistension;
public javax.swing.JComboBox jcbEstadoSensorio;
public javax.swing.JComboBox jcbGenitalesSecreciones;
public javax.swing.JComboBox jcbHidratacion;
public javax.swing.JComboBox jcbInguinales1;
public javax.swing.JComboBox jcbInguinales2;
public javax.swing.JComboBox jcbInguinales3;
public javax.swing.JComboBox jcbInguinales4;
public javax.swing.JComboBox jcbLLCap;
public javax.swing.JComboBox jcbLesion;
public javax.swing.JComboBox jcbLesionEctoparasitos;
public javax.swing.JComboBox jcbLesionPrurito;
public javax.swing.JComboBox jcbLesionSimetria;
public javax.swing.JComboBox jcbMaBucal;
public javax.swing.JComboBox jcbMaGenital;
public javax.swing.JComboBox jcbMaNasal;
public javax.swing.JComboBox jcbMaOcular;
public javax.swing.JComboBox jcbNerviosoUbicacion;
public javax.swing.JComboBox jcbParalisis;
```

```
public javax.swing.JComboBox jcbParesia;  
public javax.swing.JComboBox jcbPopliteos1;  
public javax.swing.JComboBox jcbPopliteos2;  
public javax.swing.JComboBox jcbPopliteos3;  
public javax.swing.JComboBox jcbPopliteos4;  
public javax.swing.JComboBox jcbPulsoIntensidad;  
public javax.swing.JComboBox jcbPulsoRitmo;  
public javax.swing.JComboBox jcbRegurgitacion;  
public javax.swing.JComboBox jcbRitmoCorazon;  
public javax.swing.JComboBox jcbRitmoRespiratorio;  
public javax.swing.JComboBox jcbRuidos;  
public javax.swing.JComboBox jcbSNMI;  
public javax.swing.JComboBox jcbSNMS;  
public javax.swing.JComboBox jcbSecreciones;  
public javax.swing.JComboBox jcbSensibilidad;  
public javax.swing.JComboBox jcbSubmandibular1;  
public javax.swing.JComboBox jcbSubmandibular2;  
public javax.swing.JComboBox jcbSubmandibular3;  
public javax.swing.JComboBox jcbSubmandibular4;  
public javax.swing.JComboBox jcbTipo;  
public javax.swing.JComboBox jcbTosReflejo;  
public javax.swing.JComboBox jcbVomito;  
public static javax.swing.JLabel jlAlumno;  
public static javax.swing.JLabel jlAlumno1;  
public static javax.swing.JLabel jlAlumno2;  
public static javax.swing.JLabel jlApellido;  
public javax.swing.JLabel jlCasoClinico;  
public static javax.swing.JLabel jlColor;  
public javax.swing.JLabel jlDato;  
public static javax.swing.JLabel jlDni;  
public static javax.swing.JLabel jlDocente;  
public static javax.swing.JLabel jlDocente1;  
public static javax.swing.JLabel jlDocente2;  
public static javax.swing.JLabel jlDomicilio;  
public static javax.swing.JLabel jlEspecie;  
public javax.swing.JLabel jlFecha;  
public static javax.swing.JLabel jlFechaNacimiento;  
public static javax.swing.JLabel jlNombre;  
public static javax.swing.JLabel jlPeso;  
public static javax.swing.JLabel jlRaza;  
public static javax.swing.JLabel jlSexo;  
public static javax.swing.JLabel jlTelefono;  
public javax.swing.JPanel jpAxilar;  
public javax.swing.JPanel jpClaudicacion;  
public javax.swing.JPanel jpConsulta1;  
public javax.swing.JPanel jpConsulta10;  
public javax.swing.JPanel jpConsulta2;  
public javax.swing.JPanel jpConsulta3;  
public javax.swing.JPanel jpConsulta4;  
public javax.swing.JPanel jpConsulta5;  
public javax.swing.JPanel jpConsulta6;  
public javax.swing.JPanel jpConsulta7;  
public javax.swing.JPanel jpConsulta8;  
public javax.swing.JPanel jpConsulta9;  
public javax.swing.JPanel jpConvulsion;  
public javax.swing.JPanel jpDeshidratado;  
public static javax.swing.JPanel jpGeneral;  
public javax.swing.JPanel jpGenitalesSecreciones;  
public javax.swing.JPanel jpInguinales;  
public javax.swing.JPanel jpLesion;  
public javax.swing.JPanel jpParalisis;  
public javax.swing.JPanel jpParesia;
```

```
public javax.swing.JPanel jpPopliteos;
public javax.swing.JPanel jpSecreciones;
public javax.swing.JPanel jpSubmandibular;
public javax.swing.JTextArea jtaAnamnesis;
public javax.swing.JTextArea jtaAuscultacion;
public javax.swing.JTextArea jtaBoca;
public javax.swing.JTextArea jtaCodExamenes;
public javax.swing.JTextArea jtaCorazonAuscultacion;
public javax.swing.JTextArea jtaDiagnosticoDiferencial;
public javax.swing.JTextArea jtaDiagnosticoPresuntivo;
public javax.swing.JTextArea jtaDiagnosticoPronostico;
public javax.swing.JTextArea jtaEsofago;
public javax.swing.JTextArea jtaEstomago;
public javax.swing.JTextArea jtaGenitalesExterno;
public javax.swing.JTextArea jtaGenitalesInterno;
public javax.swing.JTextArea jtaGenitalesSecreciones;
public javax.swing.JTextArea jtaHigado;
public javax.swing.JTextArea jtaIndicaciones;
public javax.swing.JTextArea jtaInstetinos;
public javax.swing.JTextArea jtaLesionManto;
public javax.swing.JTextArea jtaLesionOidos;
public javax.swing.JTextArea jtaLesionOlor;
public javax.swing.JTextArea jtaLesionTipo;
public javax.swing.JTextArea jtaLocomotorDesformacion;
public javax.swing.JTextArea jtaLocomotorLesion;
public javax.swing.JTextArea jtaLocomotorUbicacionD;
public javax.swing.JTextArea jtaLocomotorUbicacionL;
public javax.swing.JTextArea jtaMotivo;
public javax.swing.JTextArea jtaOrina;
public javax.swing.JTextArea jtaRiñones;
public javax.swing.JTextArea jtaSecreciones;
public javax.swing.JTextArea jtaTactoRectal;
public javax.swing.JTextArea jtaTratamiento;
public javax.swing.JTextArea jtaUreteres;
public javax.swing.JTextArea jtaUretra;
public javax.swing.JTextArea jtaVejiga;
public javax.swing.JTextArea jtaViasInferiores;
public javax.swing.JTextArea jtaViasSuperiores;
public javax.swing.JTextField jtfConvulsion;
public javax.swing.JTextField jtfDeshidratado;
public javax.swing.JTextField jtfFC;
public javax.swing.JTextField jtfFR;
public javax.swing.JTextField jtfLesionForma;
public javax.swing.JTextField jtfOjoDerecho;
public javax.swing.JTextField jtfOjoIzquierdo;
public javax.swing.JTextField jtfParalisis;
public javax.swing.JTextField jtfParesia;
public javax.swing.JTextField jtfReflejos;
public javax.swing.JTextField jtfTemperatura;
public javax.swing.JTabbedPane jtpPestañas;
// End of variables declaration
}
```

MostrarDerivacionAnimalG.Java

```

package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import java.sql.Date;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
public class MostrarDerivacionAnimalG extends javax.swing.JInternalFrame {
    public MostrarDerivacionAnimalG() {
        initComponents();
        Calendar Cal= Calendar.getInstance();
        String ccc =ListadoMostrarRetornoAG.auxenviar.getText();
        String r = ListadoMostrarRetornoAG.auxenviar2.getText();
        jFecha.setText(animal.GetData("fecha","SELECT `fecha` FROM derivacion WHERE cod_derivaciong = \"'+r+'\""));
        int Max= Integer.parseInt(animal.GetData("cod_derivacion", "SELECT count(cod_derivacion)AS cod_derivacion FROM derivaciong"));
        String cod_derivacion= Integer.toString(Max+1);
        jIcodigoderivacion.setText(cod_derivacion+"/"+Cal.get(Cal.YEAR));
        int cc=Integer.parseInt(ccc);
        jNombre.setText(animal.GetData("nombre","SELECT nombre FROM animal_grandes WHERE cod_animal = \"'+cc+'\""));
        int especie=Integer.parseInt(animal.GetData("cod_especie","SELECT cod_especie FROM animal_grandes WHERE cod_animal = \"'+cc+'\""));
        jEspecie.setText(animal.GetData("especie","SELECT especie FROM lista_especieg WHERE cod_especie = \"'+especie+'\""));
        int raza=Integer.parseInt(animal.GetData("cod_raza","SELECT cod_raza FROM animal_grandes WHERE cod_animal = \"'+cc+'\""));
        jRaza.setText(animal.GetData("raza","SELECT raza FROM lista_razag WHERE cod_raza = \"'+raza+'\""));
        jPeso.setText(animal.GetData("peso","SELECT peso FROM animal_grandes WHERE cod_animal = \"'+cc+'\""));
        jFechaNacimiento.setText(animal.GetData("nacimiento","SELECT nacimiento FROM animal_grandes WHERE cod_animal = \"'+cc+'\""));
        jSexo.setText(animal.GetData("sexo","SELECT sexo FROM animal_grandes WHERE cod_animal = \"'+cc+'\""));
        jColor.setText(animal.GetData("color","SELECT color FROM animal_grandes WHERE cod_animal = \"'+cc+'\""));
        int duenio=Integer.parseInt(animal.GetData("dni_duenio","SELECT dni_duenio FROM animal_grandes WHERE cod_animal = \"'+cc+'\""));
        jDni.setText(animal.GetData("dni_duenio","SELECT dni_duenio FROM animal_grandes WHERE cod_animal = \"'+cc+'\""));
        jApellido.setText(animal.GetData("apellido","SELECT apellido FROM duenio WHERE dni_duenio = \"'+duenio+'\""));
        jDomicilio.setText(animal.GetData("domicilio","SELECT domicilio FROM duenio WHERE dni_duenio = \"'+duenio+'\""));
        jTelefono.setText(animal.GetData("telefono_fijo","SELECT telefono_fijo FROM duenio WHERE dni_duenio = \"'+duenio+'\""));
        jDato.setText(animal.GetData("email","SELECT email FROM duenio WHERE dni_duenio = \"'+duenio+'\""));
        int dni_doctor=Integer.parseInt(animal.GetData("dni_doctor","SELECT dni_doctor FROM cod_derivaciong WHERE cod_derivacion = \"'+r+'\""));
        jIDocente2.setText(String.valueOf(dni_doctor));
        jIDocente.setText(animal.GetData("apellido","SELECT apellido FROM doctor WHERE dni_doctor = \"'+dni_doctor+'\""));
        jIDocente1.setText(animal.GetData("nombre","SELECT nombre FROM doctor WHERE dni_doctor = \"'+dni_doctor+'\""));
        int dni_alumno=Integer.parseInt(animal.GetData("dni_alumno","SELECT dni_alumno FROM derivacion WHERE cod_derivacion = \"'+r+'\""));
        jIAlumno2.setText(String.valueOf(dni_doctor));
        jIAlumno.setText(animal.GetData("apellido","SELECT apellido FROM alumno WHERE dni_alumno = \"'+dni_alumno+'\""));
        jIAlumno1.setText(animal.GetData("nombre","SELECT nombre FROM alumno WHERE dni_alumno = \"'+dni_alumno+'\""));
        jtaAnamnesis.setText(animal.GetData("solicitud","SELECT solicitud FROM derivaciong WHERE cod_derivacion = \"'+r+'\""));
        jtaSistema.setText(animal.GetData("sistema","SELECT sistema FROM derivaciong WHERE cod_derivacion = \"'+r+'\""));
        jtaIndicaciones.setText(animal.GetData("indicaciones","SELECT indicaciones FROM derivaciong WHERE cod_derivacion = \"'+r+'\""));
    }
    /*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")
    private void jbGuardarActionPerformed(java.awt.event.ActionEvent evt) {
        String cod_derivacion = jIcodigoderivacion.getText();
        String cod_animal=ListadoMostrarRetornoAG.auxenviar.getText();
    }

```

```

        String dni_doctor=jlDocente2.getText();
        String dni_alumno=jlAlumno2.getText();
        boolean aux = animal.existe("derivacion", "cod_derivacion", cod_derivacion);
        if (aux==true){
            JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
        }else{
            String anamnesis = jtaAnamnesis.getText();
            String sistema = jtaSistema.getText();
            String indicaciones = jtaIndicaciones.getText();
            java.util.Date date = new java.util.Date();
            java.text.SimpleDateFormat sdf=new java.text.SimpleDateFormat("yyyy-MM-dd");
            String fecha = sdf.format(date);
            String datos[] = {cod_derivacion,fecha,cod_animal,dni_doctor,dni_alumno,anamnesis,sistema,indicaciones};
            animal.InsertarRegistro(datos, "insert into derivacion values(?,?,?,?,?,?,?,?)");
        }
    }
    private void jcbDatosDueñoActionPerformed(java.awt.event.ActionEvent evt) {
        String dni= jlDni.getText();
        String dato = jcbDatosDueño.getSelectedItem().toString();
        if(dato=="e-mail"){
            jlDato.setText(animal.GetData("email","SELECT email FROM duenio WHERE dni_duenio = '"+dni+"'"));
        }
        if(dato=="facebook"){
            jlDato.setText(animal.GetData("facebook","SELECT facebook FROM duenio WHERE dni_duenio = '"+dni+"'"));
        }
        if(dato=="celular"){
            jlDato.setText(animal.GetData("telefono_celular","SELECT telefono_celular FROM duenio WHERE dni_duenio = '"+dni+"'"));
        }
    }
    MySQL animal = new MySQL();
    // Variables declaration - do not modify
    public javax.swing.JLabel jLabel1;
    public javax.swing.JLabel jLabel10;
    public javax.swing.JLabel jLabel11;
    public javax.swing.JLabel jLabel116;
    public javax.swing.JLabel jLabel12;
    public javax.swing.JLabel jLabel123;
    public javax.swing.JLabel jLabel124;
    public javax.swing.JLabel jLabel125;
    public javax.swing.JLabel jLabel126;
    public javax.swing.JLabel jLabel13;
    public javax.swing.JLabel jLabel14;
    public javax.swing.JLabel jLabel143;
    public javax.swing.JLabel jLabel15;
    public static javax.swing.JLabel jLabel16;
    public javax.swing.JLabel jLabel18;
    public javax.swing.JLabel jLabel3;
    public javax.swing.JLabel jLabel4;
    public javax.swing.JLabel jLabel5;
    public javax.swing.JLabel jLabel6;
    public javax.swing.JLabel jLabel7;
    public javax.swing.JLabel jLabel8;
    public javax.swing.JLabel jLabel9;
    public javax.swing.JScrollPane jScrollPane1;
    public javax.swing.JScrollPane jScrollPane2;
    public javax.swing.JScrollPane jScrollPane35;
    public static javax.swing.JScrollPane jScrollPane37;
    public static javax.swing.JTabbedPane jTabbedPane1;
    public javax.swing.JButton jbGuardar;
    public javax.swing.JButton jbSalir;
    public javax.swing.JComboBox jcbDatosDueño;
    public static javax.swing.JLabel jlAlumno;

```

```
public static javax.swing.JLabel jlAlumno1;
public static javax.swing.JLabel jlAlumno2;
public static javax.swing.JLabel jlApellido;
public static javax.swing.JLabel jlColor;
public javax.swing.JLabel jlDato;
public static javax.swing.JLabel jlDni;
public static javax.swing.JLabel jlDocente;
public static javax.swing.JLabel jlDocente1;
public static javax.swing.JLabel jlDocente2;
public static javax.swing.JLabel jlDomicilio;
public static javax.swing.JLabel jlEspecie;
public javax.swing.JLabel jlFecha;
public static javax.swing.JLabel jlFechaNacimiento;
public static javax.swing.JLabel jlNombre;
public static javax.swing.JLabel jlPeso;
public static javax.swing.JLabel jlRaza;
public static javax.swing.JLabel jlSexo;
public static javax.swing.JLabel jlTelefono;
public javax.swing.JLabel jlCodigoDerivacion;
public javax.swing.JPanel jpConsulta10;
public javax.swing.JPanel jpConvulsion;
public static javax.swing.JPanel jpGeneral;
public javax.swing.JPanel jpParalisis;
public javax.swing.JPanel jpParesia;
public javax.swing.JPanel jpRetorno1;
public javax.swing.JPanel jpRetorno3;
public javax.swing.JTextArea jtaAnamnesis;
public javax.swing.JTextArea jtaIndicaciones;
public javax.swing.JTextArea jtaSistema;
public javax.swing.JTabbedPane jtpPestañas;
// End of variables declaration
}
```


MostrarDerivacionAnimalP.Java

```

package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import java.sql.Date;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
public class MostrarDerivacionAnimalP extends javax.swing.JInternalFrame {
    public MostrarDerivacionAnimalP() {
        initComponents();
        Calendar Cal= Calendar.getInstance();
        String cc =ListadoDerivacionP.auxenviar.getText();
        String r = ListadoDerivacionP.auxenviar2.getText();
        jlFecha.setText(animal.GetData("fecha","SELECT `fecha` FROM derivacion WHERE cod_derivacion = \"'+r+'\""));
        int Max= Integer.parseInt(animal.GetData("cod_derivacion", "SELECT count(cod_derivacion)AS cod_derivacion FROM derivacion"));
        String cod_derivacion= Integer.toString(Max+1);
        jlcodigoderivacion.setText(""+cod_derivacion+"/"+Cal.get(Cal.YEAR));
        jlNombre.setText(animal.GetData("nombre","SELECT nombre FROM animal_pequenos WHERE cod_animal = \"'+cc+'\""));
        int especie=Integer.parseInt(animal.GetData("cod_especie","SELECT cod_especie FROM animal_pequenos WHERE cod_animal = \"'+cc+'\""));
        jlEspecie.setText(animal.GetData("especie","SELECT especie FROM lista_especie WHERE cod_especie = \"'+especie+'\""));
        int raza=Integer.parseInt(animal.GetData("cod_raza","SELECT cod_raza FROM animal_pequenos WHERE cod_animal = \"'+cc+'\""));
        jlRaza.setText(animal.GetData("raza","SELECT raza FROM lista_raza WHERE cod_raza = \"'+raza+'\""));
        jlPeso.setText(animal.GetData("peso","SELECT peso FROM animal_pequenos WHERE cod_animal = \"'+cc+'\""));
        jlFechaNacimiento.setText(animal.GetData("nacimiento","SELECT nacimiento FROM animal_pequenos WHERE cod_animal = \"'+cc+'\""));
        jlSexo.setText(animal.GetData("sexo","SELECT sexo FROM animal_pequenos WHERE cod_animal = \"'+cc+'\""));
        jlColor.setText(animal.GetData("color","SELECT color FROM animal_pequenos WHERE cod_animal = \"'+cc+'\""));
        int duenio=Integer.parseInt(animal.GetData("dni_duenio","SELECT dni_duenio FROM animal_pequenos WHERE cod_animal = \"'+cc+'\""));
        jlDni.setText(animal.GetData("dni_duenio","SELECT dni_duenio FROM animal_pequenos WHERE cod_animal = \"'+cc+'\""));
        jlApellido.setText(animal.GetData("apellido","SELECT apellido FROM duenio WHERE dni_duenio = \"'+duenio+'\""));
        jlDomicilio.setText(animal.GetData("domicilio","SELECT domicilio FROM duenio WHERE dni_duenio = \"'+duenio+'\""));
        jlTelefono.setText(animal.GetData("telefono_fijo","SELECT telefono_fijo FROM duenio WHERE dni_duenio = \"'+duenio+'\""));
        jlDato.setText(animal.GetData("email","SELECT email FROM duenio WHERE dni_duenio = \"'+duenio+'\""));
        int dni_doctor=Integer.parseInt(animal.GetData("dni_doctor","SELECT dni_doctor FROM derivacion WHERE cod_derivacion = \"'+r+'\""));
        jlDocente2.setText(String.valueOf(dni_doctor));
        jlDocente.setText(animal.GetData("apellido","SELECT apellido FROM doctor WHERE dni_doctor = \"'+dni_doctor+'\""));
        jlDocente1.setText(animal.GetData("nombre","SELECT nombre FROM doctor WHERE dni_doctor = \"'+dni_doctor+'\""));
        int dni_alumno=Integer.parseInt(animal.GetData("dni_alumno","SELECT dni_alumno FROM derivacion WHERE cod_derivacion = \"'+r+'\""));
        jlAlumno2.setText(String.valueOf(dni_doctor));
        jlAlumno.setText(animal.GetData("apellido","SELECT apellido FROM alumno WHERE dni_alumno = \"'+dni_alumno+'\""));
        jlAlumno1.setText(animal.GetData("nombre","SELECT nombre FROM alumno WHERE dni_alumno = \"'+dni_alumno+'\""));
        jtaAnamnesis.setText(animal.GetData("solicitud","SELECT solicitud FROM derivacion WHERE cod_derivacion = \"'+r+'\""));
        jtaSistema.setText(animal.GetData("sistema","SELECT sistema FROM derivacion WHERE cod_derivacion = \"'+r+'\""));
        jtaIndicaciones.setText(animal.GetData("sugerencia","SELECT sugerencia FROM derivacion WHERE cod_derivacion = \"'+r+'\""));
    }
    /* 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")
    private void jbGuardarActionPerformed(java.awt.event.ActionEvent evt) {
        String cod_derivacion = jlcodigoderivacion.getText();
        String cod_animal=ListadoDerivacionP.auxenviar.getText();
    }

```



```

String dni_doctor=jlDocente2.getText();
String dni_alumno=jlAlumno2.getText();
boolean aux = animal.existe("derivacion", "cod_derivacion", cod_derivacion);
if (aux==true){
    JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
}else{
    String anamnesis = jtaAnamnesis.getText();
    String sistema = jtaSistema.getText();
    String indicaciones = jtaIndicaciones.getText();
    java.util.Date date = new java.util.Date();
    java.text.SimpleDateFormat sdf=new java.text.SimpleDateFormat("yyyy-MM-dd");
    String fecha = sdf.format(date);
    String datos[] = {cod_derivacion,fecha,cod_animal,dni_doctor,dni_alumno,anamnesis,sistema,indicaciones};
    animal.InsertarRegistro(datos, "insert into derivacion values(?,?,?,?,?,?,?,?)");
}
}
private void jcbDatosDueñoActionPerformed(java.awt.event.ActionEvent evt) {
    String dni= jlDni.getText();
    String dato = jcbDatosDueño.getSelectedItem().toString();
    if(dato=="e-mail"){
        jlDato.setText(animal.GetData("email","SELECT email FROM duenio WHERE dni_duenio = '"+dni+"'"));
    }
    if(dato=="facebook"){
        jlDato.setText(animal.GetData("facebook","SELECT facebook FROM duenio WHERE dni_duenio = '"+dni+"'"));
    }
    if(dato=="celular"){
        jlDato.setText(animal.GetData("telefono_celular","SELECT telefono_celular FROM duenio WHERE dni_duenio = '"+dni+"'"));
    }
}
MySQL animal = new MySQL();
// Variables declaration - do not modify
public javax.swing.JLabel jLabel1;
public javax.swing.JLabel jLabel10;
public javax.swing.JLabel jLabel11;
public javax.swing.JLabel jLabel116;
public javax.swing.JLabel jLabel12;
public javax.swing.JLabel jLabel123;
public javax.swing.JLabel jLabel124;
public javax.swing.JLabel jLabel125;
public javax.swing.JLabel jLabel126;
public javax.swing.JLabel jLabel13;
public javax.swing.JLabel jLabel14;
public javax.swing.JLabel jLabel143;
public javax.swing.JLabel jLabel15;
public static javax.swing.JLabel jLabel16;
public javax.swing.JLabel jLabel18;
public javax.swing.JLabel jLabel3;
public javax.swing.JLabel jLabel4;
public javax.swing.JLabel jLabel5;
public javax.swing.JLabel jLabel6;
public javax.swing.JLabel jLabel7;
public javax.swing.JLabel jLabel8;
public javax.swing.JLabel jLabel9;
public javax.swing.JScrollPane jScrollPane1;
public javax.swing.JScrollPane jScrollPane2;
public javax.swing.JScrollPane jScrollPane35;
public static javax.swing.JScrollPane jScrollPane37;
public static javax.swing.JTabbedPane jTabbedPane1;
public javax.swing.JButton jbGuardar;
public javax.swing.JButton jbSalir;
public javax.swing.JComboBox jcbDatosDueño;
public static javax.swing.JLabel jlAlumno;

```

```
public static javax.swing.JLabel jlAlumno1;
public static javax.swing.JLabel jlAlumno2;
public static javax.swing.JLabel jlApellido;
public static javax.swing.JLabel jlColor;
public javax.swing.JLabel jlDato;
public static javax.swing.JLabel jlDni;
public static javax.swing.JLabel jlDocente;
public static javax.swing.JLabel jlDocente1;
public static javax.swing.JLabel jlDocente2;
public static javax.swing.JLabel jlDomicilio;
public static javax.swing.JLabel jlEspecie;
public javax.swing.JLabel jlFecha;
public static javax.swing.JLabel jlFechaNacimiento;
public static javax.swing.JLabel jlNombre;
public static javax.swing.JLabel jlPeso;
public static javax.swing.JLabel jlRaza;
public static javax.swing.JLabel jlSexo;
public static javax.swing.JLabel jlTelefono;
public javax.swing.JLabel jlCodigoDerivacion;
public javax.swing.JPanel jpConsulta10;
public javax.swing.JPanel jpConvulsion;
public static javax.swing.JPanel jpGeneral;
public javax.swing.JPanel jpParalisis;
public javax.swing.JPanel jpParesia;
public javax.swing.JPanel jpRetorno1;
public javax.swing.JPanel jpRetorno3;
public javax.swing.JTextArea jtaAnamnesis;
public javax.swing.JTextArea jtaIndicaciones;
public javax.swing.JTextArea jtaSistema;
public javax.swing.JTabbedPane jtpPestañas;
// End of variables declaration
}
```

MostrarRetornoAnimalG.Java

```

package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import java.sql.Date;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
public class MostrarRetornoAnimalG extends javax.swing.JInternalFrame {
    public MostrarRetornoAnimalG() {
        initComponents();
        Calendar Cal= Calendar.getInstance();
        String cc =ListadoMostrarRetornoAG.auxenviar.getText();
        String r = ListadoMostrarRetornoAG.auxenviar2.getText();
        jlFecha.setText(animal.GetData("fecha","SELECT `fecha` FROM retornog WHERE cod_retorno = \"'+r+'\""));
        jlCasoClinico.setText("\"'+cc+'\"/\"'+Cal.get(Cal.YEAR)\"");
        int cod_animal=Integer.parseInt(animal.GetData("cod_animal","SELECT cod_animal FROM consultag WHERE caso_clinico = \"'+cc+'\""));
        jlNombre.setText(animal.GetData("nombre","SELECT nombre FROM animal_grandes WHERE cod_animal = \"'+cod_animal+'\""));
        int especie=Integer.parseInt(animal.GetData("cod_especie","SELECT cod_especie FROM animal_grandes WHERE cod_animal =
        \"'+cod_animal+'\""));
        jlEspecie.setText(animal.GetData("especie","SELECT especie FROM lista_especieg WHERE cod_especie = \"'+especie+'\""));
        int raza=Integer.parseInt(animal.GetData("cod_raza","SELECT cod_raza FROM animal_grandes WHERE cod_animal = \"'+cod_animal+'\""));
        jlRaza.setText(animal.GetData("raza","SELECT raza FROM lista_razag WHERE cod_raza = \"'+raza+'\""));
        jlPeso.setText(animal.GetData("peso","SELECT peso FROM animal_grandes WHERE cod_animal = \"'+cod_animal+'\""));
        jlFechaNacimiento.setText(animal.GetData("nacimiento","SELECT nacimiento FROM animal_grandes WHERE cod_animal =
        \"'+cod_animal+'\""));
        jlSexo.setText(animal.GetData("sexo","SELECT sexo FROM animal_grandes WHERE cod_animal = \"'+cod_animal+'\""));
        jlColor.setText(animal.GetData("color","SELECT color FROM animal_grandes WHERE cod_animal = \"'+cod_animal+'\""));
        int duenio=Integer.parseInt(animal.GetData("dni_duenio","SELECT dni_duenio FROM animal_grandes WHERE cod_animal =
        \"'+cod_animal+'\""));
        jlDni.setText(animal.GetData("dni_duenio","SELECT dni_duenio FROM animal_grandes WHERE cod_animal = \"'+cod_animal+'\""));
        jlApellido.setText(animal.GetData("apellido","SELECT apellido FROM duenio WHERE dni_duenio = \"'+duenio+'\""));
        jlDomicilio.setText(animal.GetData("domicilio","SELECT domicilio FROM duenio WHERE dni_duenio = \"'+duenio+'\""));
        jlTelefono.setText(animal.GetData("telefono_fijo","SELECT telefono_fijo FROM duenio WHERE dni_duenio = \"'+duenio+'\""));
        jIDato.setText(animal.GetData("email","SELECT email FROM duenio WHERE dni_duenio = \"'+duenio+'\""));
        int dni_doctor=Integer.parseInt(animal.GetData("dni_doctor","SELECT dni_doctor FROM retornog WHERE cod_retorno = \"'+r+'\""));
        jlDocente2.setText(String.valueOf(dni_doctor));
        jlDocente.setText(animal.GetData("apellido","SELECT apellido FROM doctor WHERE dni_doctor = \"'+dni_doctor+'\""));
        jlDocente1.setText(animal.GetData("nombre","SELECT nombre FROM doctor WHERE dni_doctor = \"'+dni_doctor+'\""));
        int dni_alumno=Integer.parseInt(animal.GetData("dni_alumno","SELECT dni_alumno FROM retorno WHERE cod_retorno = \"'+r+'\""));
        jlAlumno2.setText(String.valueOf(dni_doctor));
        jlAlumno.setText(animal.GetData("apellido","SELECT apellido FROM alumno WHERE dni_alumno = \"'+dni_alumno+'\""));
        jlAlumno1.setText(animal.GetData("nombre","SELECT nombre FROM alumno WHERE dni_alumno = \"'+dni_alumno+'\""));
        jtaAnamnesis.setText(animal.GetData("anamnesis","SELECT anamnesis FROM retornog WHERE cod_retorno = \"'+r+'\""));
        jtfTemperatura.setText(animal.GetData("temperatura","SELECT temperatura FROM retornog WHERE cod_retorno = \"'+r+'\""));
        jtfFC.setText(animal.GetData("fc","SELECT fc FROM retornog WHERE cod_retorno = \"'+r+'\""));
        jtfFR.setText(animal.GetData("fr","SELECT fr FROM retornog WHERE cod_retorno = \"'+r+'\""));
        jcbCC.setSelectedItem(animal.GetData("cc","SELECT cc FROM retorno WHERE cod_retorno = \"'+r+'\""));
        jcbLLCap.setSelectedItem(animal.GetData("ll.cap","SELECT `ll.cap` FROM retornog WHERE cod_retorno = \"'+r+'\""));
        jcbPulsoIntensidad.setSelectedItem(animal.GetData("pulso_intensidad","SELECT pulso_intensidad FROM retornog WHERE cod_retorno =
        \"'+r+'\""));
        jcbPulsoRitmo.setSelectedItem(animal.GetData("pulso_Ritmo","SELECT pulso_Ritmo FROM retornog WHERE cod_retorno = \"'+r+'\""));
        String hidratacion2="Normal";
    }
}

```

```

String hidratacion=(animal.GetData("hidratacion","SELECT hidratacion FROM retornog WHERE cod_retorno = \''+r+'\'");
if (hidratacion2.equals(hidratacion)){
    jcbHidratacion.setSelectedItem(hidratacion);
}else{
    jtfDeshidratado.setText(hidratacion);
    jcbHidratacion.setSelectedItem("Deshidratado");
}
jcbMaOcular.setSelectedItem(animal.GetData("ma_ocular","SELECT ma_ocular FROM retornog WHERE cod_retorno = \''+r+'\'");
jcbMaBucal.setSelectedItem(animal.GetData("ma_bucal","SELECT ma_bucal FROM retornog WHERE cod_retorno = \''+r+'\'");
jcbMaNasal.setSelectedItem(animal.GetData("ma_nasal","SELECT ma_nasal FROM retornog WHERE cod_retorno = \''+r+'\'");
jcbMaGenital.setSelectedItem(animal.GetData("ma_genital","SELECT ma_genital FROM retornog WHERE cod_retorno = \''+r+'\'");
String submandibular = (animal.GetData("submandibular","SELECT submandibular FROM retornog WHERE cod_retorno = \''+r+'\'");
String submandibular2 = "Si";
String delimitadores = ",";
String[] palabrasSeparadas = submandibular.split(delimitadores);
if(submandibular2.equals(submandibular)){
}else{
    jcbSubmandibular1.setSelectedItem("No");
    jcbSubmandibular2.setSelectedItem(palabrasSeparadas[0]);
    jcbSubmandibular3.setSelectedItem(palabrasSeparadas[1]);
    jcbSubmandibular4.setSelectedItem(palabrasSeparadas[2]);
}
String axilar = (animal.GetData("axilar","SELECT axilar FROM retornog WHERE cod_retorno = \''+r+'\'");
String axilar2 = "Si";
if(axilar.equals(axilar2)){
}else{
    jcbAxilar1.setSelectedItem("No");
    palabrasSeparadas = axilar.split(delimitadores);
    jcbAxilar2.setSelectedItem(palabrasSeparadas[0]);
    jcbAxilar3.setSelectedItem(palabrasSeparadas[1]);
    jcbAxilar4.setSelectedItem(palabrasSeparadas[2]);
}
String inguinales = (animal.GetData("inguinales","SELECT inguinales FROM retornog WHERE cod_retorno = \''+r+'\'");
String inguinales2 = "Si";
if(inguinales.equals(inguinales2)){
}else{
    jcbSubmandibular1.setSelectedItem("No");
    palabrasSeparadas = inguinales.split(delimitadores);
    jcbInguinales2.setSelectedItem(palabrasSeparadas[0]);
    jcbInguinales3.setSelectedItem(palabrasSeparadas[1]);
    jcbInguinales4.setSelectedItem(palabrasSeparadas[2]);
}
String popliteos = (animal.GetData("popliteos","SELECT popliteos FROM retornog WHERE cod_retorno = \''+r+'\'");
String popliteos2 = "Si";
if(popliteos.equals(popliteos2)){
}else{
    jcbPopliteos1.setSelectedItem("No");
    palabrasSeparadas = popliteos.split(delimitadores);
    jcbPopliteos2.setSelectedItem(palabrasSeparadas[0]);
    jcbPopliteos3.setSelectedItem(palabrasSeparadas[1]);
    jcbPopliteos4.setSelectedItem(palabrasSeparadas[2]);
}
jtaSistema.setText(animal.GetData("sistema","SELECT sistema FROM retornog WHERE cod_retorno = \''+r+'\'");
jtaIndicaciones.setText(animal.GetData("indicaciones","SELECT indicaciones FROM retornog WHERE cod_retorno = \''+r+'\'");
}
/*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")
private void jcbHidratacionActionPerformed(java.awt.event.ActionEvent evt) {
    String hidratado="Deshidratado";
    String hidratado2=(String)jcbHidratacion.getSelectedItem();

```

```
if (hidratado2==hidratado){
    jpDeshidratado.setVisible(true);
}else{
    jpDeshidratado.setVisible(false);
}
}

private void jcbSubmandibular1ActionPerformed(java.awt.event.ActionEvent evt) {
    String submandibular = (String)jcbSubmandibular1.getSelectedItem();
    String submandibular2 = "No";
    if(submandibular==submandibular2){
        jpSubmandibular.setVisible(true);
    }else{
        jpSubmandibular.setVisible(false);
    }
}

private void jcbAxilar1ActionPerformed(java.awt.event.ActionEvent evt) {
    String axilar = (String)jcbAxilar1.getSelectedItem();
    String axilar2 = "No";
    if(axilar==axilar2){
        jpAxilar.setVisible(true);
    }else{
        jpAxilar.setVisible(false);
    }
}

private void jcbInguinales1ActionPerformed(java.awt.event.ActionEvent evt) {
    String inguinales = (String)jcbInguinales1.getSelectedItem();
    String inguinales2 = "No";
    if(inguinales==inguinales2){
        jpInguinales.setVisible(true);
    }else{
        jpInguinales.setVisible(false);
    }
}

private void jcbPopliteos1ActionPerformed(java.awt.event.ActionEvent evt) {
    String popliteos = (String)jcbPopliteos1.getSelectedItem();
    String popliteos2 = "No";
    if(popliteos==popliteos2){
        jpPopliteos.setVisible(true);
    }else{
        jpPopliteos.setVisible(false);
    }
}

private void jbGuardarActionPerformed(java.awt.event.ActionEvent evt) {
    String caso_clinico = jCasoClinico.getText();
    /* Buscamos el ultimo cod_retorno y se suma 1 es decir, si el ultimo es 2 , se guarda 3
    int Max= Integer.parseInt(animal.GetData("cod_retorno", "SELECT MAX(cod_retorno) cod_retorno FROM retornog"));
    String cod_retorno= Integer.toString(Max+1);
    String cod_animal="1";
    String dni_doctor=jlDocente2.getText();
    String dni_alumno=jlAlumno2.getText();
    boolean aux = animal.existe("retornog", "cod_retorno", cod_retorno);
    if (aux==true){
        JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
    }else{
        String anamnesis = jtaAnamnesis.getText();
        String temperatura = jtfTemperatura.getText();
        String fc = jtfFC.getText();
        String fr = jtfFR.getText();
        String cc = (String)jcbCC.getSelectedItem();
        String ll_cap = (String)jcbLLCap.getSelectedItem();
        String pulso_ritmo = (String)jcbPulsoRitmo.getSelectedItem();
        String pulso_intensidad = (String)jcbPulsoIntensidad.getSelectedItem();
```

```

        String hidratacion2="Deshidratado";
String hidratacion=(String)jcbHidratacion.getSelectedItemId();
if (hidratacion2==hidratacion){
    hidratacion = jtfDeshidratado.getText();
}else{
    hidratacion=(String)jcbHidratacion.getSelectedItemId();
}
String ma_ocular = (String)jcbMaOcular.getSelectedItemId();
String ma_bucal = (String)jcbMaBucal.getSelectedItemId();
String ma_nasal = (String)jcbMaNasal.getSelectedItemId();
String ma_genital = (String)jcbMaGenital.getSelectedItemId();
String submandibular = (String)jcbSubmandibular1.getSelectedItemId();
String submandibular2 = "No";
if(submandibular==submandibular2){
    submandibular =
(String)jcbSubmandibular2.getSelectedItemId()+" "+jcbSubmandibular3.getSelectedItemId()+" "+jcbSubmandibular4.getSelectedItemId();
}else{
    submandibular = "Si";
}
String axilar = (String)jcbAxilar1.getSelectedItemId();
String axilar2 = "No";
if(axilar==axilar2){
    axilar = (String)jcbAxilar2.getSelectedItemId()+" "+jcbAxilar3.getSelectedItemId()+" "+jcbAxilar4.getSelectedItemId();
}else{
    axilar = "Si";
}
String inguinales = (String)jcbInguinales1.getSelectedItemId();
String inguinales2 = "No";
if(inguinales==inguinales2){
    inguinales = (String)jcbInguinales2.getSelectedItemId()+" "+jcbInguinales3.getSelectedItemId()+" "+jcbInguinales4.getSelectedItemId();
}else{
    inguinales = "Si";
}
String popliteos = (String)jcbPopliteos1.getSelectedItemId();
String popliteos2 = "No";
if(popliteos==popliteos2){
    popliteos = (String)jcbPopliteos2.getSelectedItemId()+" "+jcbPopliteos3.getSelectedItemId()+" "+jcbPopliteos4.getSelectedItemId();
}else{
    popliteos = "Si";
}
String sistema = jtaSistema.getText();
String indicaciones = jtaIndicaciones.getText();

java.util.Date date = new java.util.Date();
java.text.SimpleDateFormat sdf=new java.text.SimpleDateFormat("yyyy-MM-dd");
String fecha = sdf.format(date);
String datos[] =
{cod_retorno,caso_clinico,fecha,cod_animal,dni_doctor,dni_alumno,anamnesis,temperatura,fc,fr,cc,ll_cap,pulso_ritmo,pulso_intensidad,hidratacion,
ma_ocular,ma_bucal,ma_nasal,ma_genital,submandibular,axilar,inguinales,popliteos,sistema,indicaciones};    animal.InsertarRegistro(datos,
"insert into retornog values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");
}
}

private void jBuscarAlumnoActionPerformed(java.awt.event.ActionEvent evt) {
    BuscarAlumnoRetornoG buscarAlumno = new BuscarAlumnoRetornoG();
    PantallaPrincipal.jdpFondo.add(buscarAlumno);
    buscarAlumno.toFront();
    buscarAlumno.setVisible(true);
}

private void jBuscarDocenteActionPerformed(java.awt.event.ActionEvent evt) {
    BuscarDocenteRetornoG buscarDocente = new BuscarDocenteRetornoG();
    PantallaPrincipal.jdpFondo.add(buscarDocente);
    buscarDocente.toFront();
}

```



```

        buscarDocente.setVisible(true);
    }
    private void jcbDatosDueñoActionPerformed(java.awt.event.ActionEvent evt) {
        String dni= jIDni.getText();
        String dato = jcbDatosDueño.getSelectedItem().toString();
        if(dato=="e-mail"){
            jIDato.setText(animal.GetData("email","SELECT email FROM duenio WHERE dni_duenio = \''"+dni+'\'"));
        }
        if(dato=="facebook"){
            jIDato.setText(animal.GetData("facebook","SELECT facebook FROM duenio WHERE dni_duenio = \''"+dni+'\'"));
        }
        if(dato=="celular"){
            jIDato.setText(animal.GetData("telefono_celular","SELECT telefono_celular FROM duenio WHERE dni_duenio = \''"+dni+'\'"));
        }
    }
    private void jbSalirActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    MySQL animal = new MySQL();
    // Variables declaration - do not modify
    public javax.swing.JButton jBuscarAlumno;
    public javax.swing.JButton jBuscarDocente;
    public javax.swing.JLabel jLabel1;
    public javax.swing.JLabel jLabel10;
    public javax.swing.JLabel jLabel11;
    public javax.swing.JLabel jLabel116;
    public javax.swing.JLabel jLabel12;
    public javax.swing.JLabel jLabel123;
    public javax.swing.JLabel jLabel124;
    public javax.swing.JLabel jLabel125;
    public javax.swing.JLabel jLabel126;
    public javax.swing.JLabel jLabel13;
    public javax.swing.JLabel jLabel14;
    public javax.swing.JLabel jLabel143;
    public javax.swing.JLabel jLabel15;
    public static javax.swing.JLabel jLabel16;
    public javax.swing.JLabel jLabel18;
    public javax.swing.JLabel jLabel19;
    public javax.swing.JLabel jLabel2;
    public javax.swing.JLabel jLabel20;
    public javax.swing.JLabel jLabel21;
    public javax.swing.JLabel jLabel22;
    public javax.swing.JLabel jLabel23;
    public javax.swing.JLabel jLabel24;
    public javax.swing.JLabel jLabel25;
    public javax.swing.JLabel jLabel26;
    public javax.swing.JLabel jLabel27;
    public javax.swing.JLabel jLabel28;
    public javax.swing.JLabel jLabel29;
    public javax.swing.JLabel jLabel3;
    public javax.swing.JLabel jLabel30;
    public javax.swing.JLabel jLabel31;
    public javax.swing.JLabel jLabel32;
    public javax.swing.JLabel jLabel33;
    public javax.swing.JLabel jLabel34;
    public javax.swing.JLabel jLabel35;
    public javax.swing.JLabel jLabel36;
    public javax.swing.JLabel jLabel37;
    public javax.swing.JLabel jLabel38;
    public javax.swing.JLabel jLabel39;
    public javax.swing.JLabel jLabel4;
    public javax.swing.JLabel jLabel40;

```



```
public javax.swing.JLabel jLabel41;
public javax.swing.JLabel jLabel42;
public javax.swing.JLabel jLabel43;
public javax.swing.JLabel jLabel44;
public javax.swing.JLabel jLabel45;
public javax.swing.JLabel jLabel46;
public javax.swing.JLabel jLabel47;
public javax.swing.JLabel jLabel5;
public javax.swing.JLabel jLabel6;
public javax.swing.JLabel jLabel7;
public javax.swing.JLabel jLabel8;
public javax.swing.JLabel jLabel9;
public javax.swing.JScrollPane jScrollPane1;
public javax.swing.JScrollPane jScrollPane2;
public javax.swing.JScrollPane jScrollPane35;
public static javax.swing.JScrollPane jScrollPane37;
public static javax.swing.JTabbedPane jTabbedPane1;
public javax.swing.JButton jbGuardar;
public javax.swing.JButton jbSalir;
public javax.swing.JComboBox jcbAxilar1;
public javax.swing.JComboBox jcbAxilar2;
public javax.swing.JComboBox jcbAxilar3;
public javax.swing.JComboBox jcbAxilar4;
public javax.swing.JComboBox jcbCC;
public javax.swing.JComboBox jcbDatosDueño;
public javax.swing.JComboBox jcbHidratacion;
public javax.swing.JComboBox jcbInguinales1;
public javax.swing.JComboBox jcbInguinales2;
public javax.swing.JComboBox jcbInguinales3;
public javax.swing.JComboBox jcbInguinales4;
public javax.swing.JComboBox jcbLLCap;
public javax.swing.JComboBox jcbMaBucal;
public javax.swing.JComboBox jcbMaGenital;
public javax.swing.JComboBox jcbMaNasal;
public javax.swing.JComboBox jcbMaOcular;
public javax.swing.JComboBox jcbPopliteos1;
public javax.swing.JComboBox jcbPopliteos2;
public javax.swing.JComboBox jcbPopliteos3;
public javax.swing.JComboBox jcbPopliteos4;
public javax.swing.JComboBox jcbPulsoIntensidad;
public javax.swing.JComboBox jcbPulsoRitmo;
public javax.swing.JComboBox jcbSubmandibular1;
public javax.swing.JComboBox jcbSubmandibular2;
public javax.swing.JComboBox jcbSubmandibular3;
public javax.swing.JComboBox jcbSubmandibular4;
public static javax.swing.JLabel jlAlumno;
public static javax.swing.JLabel jlAlumno1;
public static javax.swing.JLabel jlAlumno2;
public static javax.swing.JLabel jlApellido;
public javax.swing.JLabel jlCasoClinico;
public static javax.swing.JLabel jlColor;
public javax.swing.JLabel jlDato;
public static javax.swing.JLabel jlDni;
public static javax.swing.JLabel jlDocente;
public static javax.swing.JLabel jlDocente1;
public static javax.swing.JLabel jlDocente2;
public static javax.swing.JLabel jlDomicilio;
public static javax.swing.JLabel jlEspecie;
public javax.swing.JLabel jlFecha;
public static javax.swing.JLabel jlFechaNacimiento;
public javax.swing.JLabel jlHora;
public static javax.swing.JLabel jlNombre;
```

```
public static javax.swing.JLabel jlPeso;  
public static javax.swing.JLabel jlRaza;  
public static javax.swing.JLabel jlSexo;  
public static javax.swing.JLabel jlTelefono;  
public javax.swing.JPanel jpAxilar;  
public javax.swing.JPanel jpConsulta10;  
public javax.swing.JPanel jpConvulsion;  
public javax.swing.JPanel jpDeshidratado;  
public static javax.swing.JPanel jpGeneral;  
public javax.swing.JPanel jpInguinales;  
public javax.swing.JPanel jpParalisis;  
public javax.swing.JPanel jpParesia;  
public javax.swing.JPanel jpPopliteos;  
public javax.swing.JPanel jpRetorno1;  
public javax.swing.JPanel jpRetorno2;  
public javax.swing.JPanel jpRetorno3;  
public javax.swing.JPanel jpSubmandibular;  
public javax.swing.JTextArea jtaAnamnesis;  
public javax.swing.JTextArea jtaIndicaciones;  
public javax.swing.JTextArea jtaSistema;  
public javax.swing.JTextField jtfDeshidratado;  
public javax.swing.JTextField jtfFC;  
public javax.swing.JTextField jtfFR;  
public javax.swing.JTextField jtfTemperatura;  
public javax.swing.JTabbedPane jtpPestañas;  
// End of variables declaration      }
```

MostrarRetornoAnimalP.Java

```

package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import java.sql.Date;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
public class MostrarRetornoAnimalP extends javax.swing.JFrame {
    public MostrarRetornoAnimalP() {
        initComponents();
        Calendar Cal= Calendar.getInstance();
        String cc =ListadoMostrarRetornoAP.auxenviar.getText();
        String r = ListadoMostrarRetornoAP.auxenviar2.getText();
        jlFecha.setText(animal.GetData("fecha","SELECT `fecha` FROM retorno WHERE cod_retorno = \"'+r+'\""));
        jlCasoClinico.setText("\"'+cc+'\"/\"'+Cal.get(Cal.YEAR)\"");
        int cod_animal=Integer.parseInt(animal.GetData("cod_animal","SELECT cod_animal FROM consulta WHERE caso_clinico = \"'+cc+'\""));
        jlNombre.setText(animal.GetData("nombre","SELECT nombre FROM animal_pequenos WHERE cod_animal = \"'+cod_animal+'\""));
        int especie=Integer.parseInt(animal.GetData("cod_especie","SELECT cod_especie FROM animal_pequenos WHERE cod_animal =
\"'+cod_animal+'\""));
        jlEspecie.setText(animal.GetData("especie","SELECT especie FROM lista_especie WHERE cod_especie = \"'+especie+'\""));
        int raza=Integer.parseInt(animal.GetData("cod_raza","SELECT cod_raza FROM animal_pequenos WHERE cod_animal = \"'+cod_animal+'\""));
        jlRaza.setText(animal.GetData("raza","SELECT raza FROM lista_raza WHERE cod_raza = \"'+raza+'\""));
        jlPeso.setText(animal.GetData("peso","SELECT peso FROM animal_pequenos WHERE cod_animal = \"'+cod_animal+'\""));
        jlFechaNacimiento.setText(animal.GetData("nacimiento","SELECT nacimiento FROM animal_pequenos WHERE cod_animal =
\"'+cod_animal+'\""));
        jlSexo.setText(animal.GetData("sexo","SELECT sexo FROM animal_pequenos WHERE cod_animal = \"'+cod_animal+'\""));
        jlColor.setText(animal.GetData("color","SELECT color FROM animal_pequenos WHERE cod_animal = \"'+cod_animal+'\""));
        int duenio=Integer.parseInt(animal.GetData("dni_duenio","SELECT dni_duenio FROM animal_pequenos WHERE cod_animal =
\"'+cod_animal+'\""));
        jlDni.setText(animal.GetData("dni_duenio","SELECT dni_duenio FROM animal_pequenos WHERE cod_animal = \"'+cod_animal+'\""));
        jlApellido.setText(animal.GetData("apellido","SELECT apellido FROM duenio WHERE dni_duenio = \"'+duenio+'\""));
        jlDomicilio.setText(animal.GetData("domicilio","SELECT domicilio FROM duenio WHERE dni_duenio = \"'+duenio+'\""));
        jlTelefono.setText(animal.GetData("telefono_fijo","SELECT telefono_fijo FROM duenio WHERE dni_duenio = \"'+duenio+'\""));
        jlDato.setText(animal.GetData("email","SELECT email FROM duenio WHERE dni_duenio = \"'+duenio+'\""));
        int dni_doctor=Integer.parseInt(animal.GetData("dni_doctor","SELECT dni_doctor FROM retorno WHERE cod_retorno = \"'+r+'\""));
        jlDocente2.setText(String.valueOf(dni_doctor));
        jlDocente.setText(animal.GetData("apellido","SELECT apellido FROM doctor WHERE dni_doctor = \"'+dni_doctor+'\""));
        jlDocente1.setText(animal.GetData("nombre","SELECT nombre FROM doctor WHERE dni_doctor = \"'+dni_doctor+'\""));
        int dni_alumno=Integer.parseInt(animal.GetData("dni_alumno","SELECT dni_alumno FROM retorno WHERE cod_retorno = \"'+r+'\""));
        jlAlumno2.setText(String.valueOf(dni_alumno));
        jlAlumno.setText(animal.GetData("apellido","SELECT apellido FROM alumno WHERE dni_alumno = \"'+dni_alumno+'\""));
        jlAlumno1.setText(animal.GetData("nombre","SELECT nombre FROM alumno WHERE dni_alumno = \"'+dni_alumno+'\""));
        jtaAnamnesis.setText(animal.GetData("anamnesis","SELECT anamnesis FROM retorno WHERE cod_retorno = \"'+r+'\""));
        jtfTemperatura.setText(animal.GetData("temperatura","SELECT temperatura FROM retorno WHERE cod_retorno = \"'+r+'\""));
        jtfFC.setText(animal.GetData("fc","SELECT fc FROM retorno WHERE cod_retorno = \"'+r+'\""));
        jtfFR.setText(animal.GetData("fr","SELECT fr FROM retorno WHERE cod_retorno = \"'+r+'\""));
        jcbCC.setSelectedItem(animal.GetData("cc","SELECT cc FROM retorno WHERE cod_retorno = \"'+r+'\""));
        jcbLLCap.setSelectedItem(animal.GetData("Il.cap","SELECT `Il.cap` FROM retorno WHERE cod_retorno = \"'+r+'\""));
        jcbPulsoIntensidad.setSelectedItem(animal.GetData("pulso_intensidad","SELECT pulso_intensidad FROM retorno WHERE cod_retorno =
\"'+r+'\""));
        jcbPulsoRitmo.setSelectedItem(animal.GetData("pulso_Ritmo","SELECT pulso_Ritmo FROM retorno WHERE cod_retorno = \"'+r+'\""));
        String hidratacion2="Normal";
    }
}

```

```

String hidratacion=(animal.GetData("hidratacion","SELECT hidratacion FROM retorno WHERE cod_retorno = \"\"+r+\"\"");
if (hidratacion2.equals(hidratacion)){
    jcbHidratacion.setSelectedItem(hidratacion);
}else{
    jtfDeshidratado.setText(hidratacion);
    jcbHidratacion.setSelectedItem("Deshidratado");
}
jcbMaOcular.setSelectedItem(animal.GetData("ma_ocular","SELECT ma_ocular FROM retorno WHERE cod_retorno = \"\"+r+\"\"");
jcbMaBucal.setSelectedItem(animal.GetData("ma_bucal","SELECT ma_bucal FROM retorno WHERE cod_retorno = \"\"+r+\"\"");
jcbMaNasal.setSelectedItem(animal.GetData("ma_nasal","SELECT ma_nasal FROM retorno WHERE cod_retorno = \"\"+r+\"\"");
jcbMaGenital.setSelectedItem(animal.GetData("ma_genital","SELECT ma_genital FROM retorno WHERE cod_retorno = \"\"+r+\"\"");
String submandibular = (animal.GetData("submandibular","SELECT submandibular FROM retorno WHERE cod_retorno = \"\"+r+\"\"");
String submandibular2 = "Si";
String delimitadores= ",";
String[] palabrasSeparadas = submandibular.split(delimitadores);
if(submandibular2.equals(submandibular)){
}else{
    jcbSubmandibular1.setSelectedItem("No");
    jcbSubmandibular2.setSelectedItem(palabrasSeparadas[0]);
    jcbSubmandibular3.setSelectedItem(palabrasSeparadas[1]);
    jcbSubmandibular4.setSelectedItem(palabrasSeparadas[2]); }
String axilar = (animal.GetData("axilar","SELECT axilar FROM retorno WHERE cod_retorno = \"\"+r+\"\"");
String axilar2 = "Si";
if(axilar.equals(axilar2)){
}else{
    jcbAxilar1.setSelectedItem("No");
    palabrasSeparadas = axilar.split(delimitadores);
    jcbAxilar2.setSelectedItem(palabrasSeparadas[0]);
    jcbAxilar3.setSelectedItem(palabrasSeparadas[1]);
    jcbAxilar4.setSelectedItem(palabrasSeparadas[2]); }
String inguinales = (animal.GetData("inguinales","SELECT inguinales FROM retorno WHERE cod_retorno = \"\"+r+\"\"");
String inguinales2 = "Si";
if(inguinales.equals(inguinales2)){
}else{
    jcbSubmandibular1.setSelectedItem("No");
    palabrasSeparadas = inguinales.split(delimitadores);
    jcbInguinales2.setSelectedItem(palabrasSeparadas[0]);
    jcbInguinales3.setSelectedItem(palabrasSeparadas[1]);
    jcbInguinales4.setSelectedItem(palabrasSeparadas[2]); }
String popliteos = (animal.GetData("popliteos","SELECT popliteos FROM retorno WHERE cod_retorno = \"\"+r+\"\"");
String popliteos2 = "Si";
if(popliteos.equals(popliteos2)){
}else{
    jcbPopliteos1.setSelectedItem("No");
    palabrasSeparadas = popliteos.split(delimitadores);
    jcbPopliteos2.setSelectedItem(palabrasSeparadas[0]);
    jcbPopliteos3.setSelectedItem(palabrasSeparadas[1]);
    jcbPopliteos4.setSelectedItem(palabrasSeparadas[2]); }
jtaSistema.setText(animal.GetData("sistema","SELECT sistema FROM retorno WHERE cod_retorno = \"\"+r+\"\"");
jtaIndicaciones.setText(animal.GetData("indicaciones","SELECT indicaciones FROM retorno WHERE cod_retorno = \"\"+r+\"\"");
}
/*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")
private void jcbHidratacionActionPerformed(java.awt.event.ActionEvent evt) {
    String hidratado="Deshidratado";
    String hidratado2=(String)jcbHidratacion.getSelectedItem();
    if (hidratado2==hidratado){
        jpDeshidratado.setVisible(true);
    }else{
        jpDeshidratado.setVisible(false);    }
}

```

```

}
private void jcbSubmandibular1ActionPerformed(java.awt.event.ActionEvent evt) {
    String submandibular = (String)jcbSubmandibular1.getSelectedItemAt();
    String submandibular2 = "No";
    if(submandibular==submandibular2){
        jpSubmandibular.setVisible(true);
    }else{
        jpSubmandibular.setVisible(false);    }
}
private void jcbAxilar1ActionPerformed(java.awt.event.ActionEvent evt) {
    String axilar = (String)jcbAxilar1.getSelectedItemAt();
    String axilar2 = "No";
    if(axilar==axilar2){
        jpAxilar.setVisible(true);
    }else{
        jpAxilar.setVisible(false);
    }
}
private void jcbInguinales1ActionPerformed(java.awt.event.ActionEvent evt) {
    String inguinales = (String)jcbInguinales1.getSelectedItemAt();
    String inguinales2 = "No";
    if(inguinales==inguinales2){
        jpInguinales.setVisible(true);
    }else{
        jpInguinales.setVisible(false);
    }
}
private void jcbPopliteos1ActionPerformed(java.awt.event.ActionEvent evt) {
    String popliteos = (String)jcbPopliteos1.getSelectedItemAt();
    String popliteos2 = "No";
    if(popliteos==popliteos2){
        jpPopliteos.setVisible(true);
    }else{
        jpPopliteos.setVisible(false);
    }
}
private void jbGuardarActionPerformed(java.awt.event.ActionEvent evt) {
    String caso_clinico = BuscarCasoClinico.auxenviar.getText();
    /* Buscamos el ultimo cod_retorno y se suma 1 es decir, si el ultimo es 2 , se guarda 3
    int Max= Integer.parseInt(animal.GetData("cod_retorno", "SELECT count(cod_retorno)AS cod_retorno FROM retorno"));
    String cod_retorno= Integer.toString(Max+1);
    String cod_animal="1";
    String dni_doctor=jlDocente2.getText();
    String dni_alumno=jlAlumno2.getText();
    boolean aux = animal.existe("retorno", "cod_retorno", cod_retorno);
    if (aux==true){
        JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
    }else{
        String anamnesis = jtaAnamnesis.getText();
        String temperatura = jtfTemperatura.getText();
        String fc = jtfFC.getText();
        String fr = jtfFR.getText();
        String cc = (String)jcbCC.getSelectedItemAt();
        String ll_cap = (String)jcbLLCap.getSelectedItemAt();
        String pulso_ritmo = (String)jcbPulsoRitmo.getSelectedItemAt();
        String pulso_intensidad = (String)jcbPulsoIntensidad.getSelectedItemAt();
        String hidratacion2="Deshidratado";
        String hidratacion=(String)jcbHidratacion.getSelectedItemAt();
        if (hidratacion2==hidratacion){
            hidratacion = jtfDeshidratado.getText();
        }else{
            hidratacion=(String)jcbHidratacion.getSelectedItemAt();

```

```

    }
    String ma_ocular = (String)jcbMaOcular.getSelectedItemAt();
    String ma_bucal = (String)jcbMaBucal.getSelectedItemAt();
    String ma_nasal = (String)jcbMaNasal.getSelectedItemAt();
    String ma_genital = (String)jcbMaGenital.getSelectedItemAt();
    String submandibular = (String)jcbSubmandibular1.getSelectedItemAt();
    String submandibular2 = "No";
    if(submandibular==submandibular2){
        submandibular =
        (String)jcbSubmandibular2.getSelectedItemAt()+" "+jcbSubmandibular3.getSelectedItemAt()+" "+jcbSubmandibular4.getSelectedItemAt();
    }else{
        submandibular = "Si";
    }
    String axilar = (String)jcbAxilar1.getSelectedItemAt();
    String axilar2 = "No";
    if(axilar==axilar2){
        axilar = (String)jcbAxilar2.getSelectedItemAt()+" "+jcbAxilar3.getSelectedItemAt()+" "+jcbAxilar4.getSelectedItemAt();
    }else{
        axilar = "Si";
    }
    String inguinales = (String)jcbInguinales1.getSelectedItemAt();
    String inguinales2 = "No";
    if(inguinales==inguinales2){
        inguinales = (String)jcbInguinales2.getSelectedItemAt()+" "+jcbInguinales3.getSelectedItemAt()+" "+jcbInguinales4.getSelectedItemAt();
    }else{
        inguinales = "Si";
    }
    String popliteos = (String)jcbPopliteos1.getSelectedItemAt();
    String popliteos2 = "No";
    if(popliteos==popliteos2){
        popliteos = (String)jcbPopliteos2.getSelectedItemAt()+" "+jcbPopliteos3.getSelectedItemAt()+" "+jcbPopliteos4.getSelectedItemAt();
    }else{
        popliteos = "Si";
    }
    String sistema = jtaSistema.getText();
    String indicaciones = jtaIndicaciones.getText();

    java.util.Date date = new java.util.Date();
    java.text.SimpleDateFormat sdf=new java.text.SimpleDateFormat("yyyy-MM-dd");
    String fecha = sdf.format(date);
    String datos[] =
    {cod_retorno,caso_clinico,fecha,cod_animal,dni_doctor,dni_alumno,anamnesis,temperatura,fc,fr,cc,Il_cap,pulso_ritmo,pulso_intensidad,hidratacion,
    ma_ocular,ma_bucal,ma_nasal,ma_genital,submandibular,axilar,inguinales,popliteos,sistema,indicaciones};
    animal.InsertarRegistro(datos, "insert into retorno values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");
    }
}

private void jBuscarAlumnoActionPerformed(java.awt.event.ActionEvent evt) {
    BuscarAlumnoRetorno buscarAlumno = new BuscarAlumnoRetorno();
    PantallaPrincipal.jdpFondo.add(buscarAlumno);
    buscarAlumno.toFront();
    buscarAlumno.setVisible(true);
}

private void jBuscarDocenteActionPerformed(java.awt.event.ActionEvent evt) {
    BuscarDocenteRetorno buscarDocente = new BuscarDocenteRetorno();
    PantallaPrincipal.jdpFondo.add(buscarDocente);
    buscarDocente.toFront();
    buscarDocente.setVisible(true);
}

private void jcbDatosDueñoActionPerformed(java.awt.event.ActionEvent evt) {
    String dni= jIdni.getText();
    String dato = jcbDatosDueño.getSelectedItemAt().toString();
    if(dato=="e-mail"){

```



```
jIDato.setText(animal.GetData("email","SELECT email FROM duenio WHERE dni_duenio = \"'+dni+'\""));
}
if(dato=="facebook"){
    jIDato.setText(animal.GetData("facebook","SELECT facebook FROM duenio WHERE dni_duenio = \"'+dni+'\""));
}
if(dato=="celular"){
    jIDato.setText(animal.GetData("telefono_celular","SELECT telefono_celular FROM duenio WHERE dni_duenio = \"'+dni+'\""));
}
}
private void jbSalirActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}
MySQL animal = new MySQL();
// Variables declaration - do not modify
public javax.swing.JButton jButtonAlumno;
public javax.swing.JButton jButtonDocente;
public javax.swing.JLabel jLabel1;
public javax.swing.JLabel jLabel10;
public javax.swing.JLabel jLabel11;
public javax.swing.JLabel jLabel116;
public javax.swing.JLabel jLabel12;
public javax.swing.JLabel jLabel123;
public javax.swing.JLabel jLabel124;
public javax.swing.JLabel jLabel125;
public javax.swing.JLabel jLabel126;
public javax.swing.JLabel jLabel13;
public javax.swing.JLabel jLabel14;
public javax.swing.JLabel jLabel143;
public javax.swing.JLabel jLabel15;
public static javax.swing.JLabel jLabel16;
public javax.swing.JLabel jLabel18;
public javax.swing.JLabel jLabel19;
public javax.swing.JLabel jLabel2;
public javax.swing.JLabel jLabel20;
public javax.swing.JLabel jLabel21;
public javax.swing.JLabel jLabel22;
public javax.swing.JLabel jLabel23;
public javax.swing.JLabel jLabel24;
public javax.swing.JLabel jLabel25;
public javax.swing.JLabel jLabel26;
public javax.swing.JLabel jLabel27;
public javax.swing.JLabel jLabel28;
public javax.swing.JLabel jLabel29;
public javax.swing.JLabel jLabel3;
public javax.swing.JLabel jLabel30;
public javax.swing.JLabel jLabel31;
public javax.swing.JLabel jLabel32;
public javax.swing.JLabel jLabel33;
public javax.swing.JLabel jLabel34;
public javax.swing.JLabel jLabel35;
public javax.swing.JLabel jLabel36;
public javax.swing.JLabel jLabel37;
public javax.swing.JLabel jLabel38;
public javax.swing.JLabel jLabel39;
public javax.swing.JLabel jLabel4;
public javax.swing.JLabel jLabel40;
public javax.swing.JLabel jLabel41;
public javax.swing.JLabel jLabel42;
public javax.swing.JLabel jLabel43;
public javax.swing.JLabel jLabel44;
public javax.swing.JLabel jLabel45;
public javax.swing.JLabel jLabel46;
```



```
public javax.swing.JLabel jLabel47;
public javax.swing.JLabel jLabel5;
public javax.swing.JLabel jLabel6;
public javax.swing.JLabel jLabel7;
public javax.swing.JLabel jLabel8;
public javax.swing.JLabel jLabel9;
public javax.swing.JScrollPane jScrollPane1;
public javax.swing.JScrollPane jScrollPane2;
public javax.swing.JScrollPane jScrollPane35;
public static javax.swing.JScrollPane jScrollPane37;
public static javax.swing.JTabbedPane jTabbedPane1;
public javax.swing.JButton jbGuardar;
public javax.swing.JButton jbSalir;
public javax.swing.JComboBox jcbAxilar1;
public javax.swing.JComboBox jcbAxilar2;
public javax.swing.JComboBox jcbAxilar3;
public javax.swing.JComboBox jcbAxilar4;
public javax.swing.JComboBox jcbCC;
public javax.swing.JComboBox jcbDatosDueño;
public javax.swing.JComboBox jcbHidratacion;
public javax.swing.JComboBox jcbInguinales1;
public javax.swing.JComboBox jcbInguinales2;
public javax.swing.JComboBox jcbInguinales3;
public javax.swing.JComboBox jcbInguinales4;
public javax.swing.JComboBox jcbLLCap;
public javax.swing.JComboBox jcbMaBucal;
public javax.swing.JComboBox jcbMaGenital;
public javax.swing.JComboBox jcbMaNasal;
public javax.swing.JComboBox jcbMaOcular;
public javax.swing.JComboBox jcbPopliteos1;
public javax.swing.JComboBox jcbPopliteos2;
public javax.swing.JComboBox jcbPopliteos3;
public javax.swing.JComboBox jcbPopliteos4;
public javax.swing.JComboBox jcbPulsoIntensidad;
public javax.swing.JComboBox jcbPulsoRitmo;
public javax.swing.JComboBox jcbSubmandibular1;
public javax.swing.JComboBox jcbSubmandibular2;
public javax.swing.JComboBox jcbSubmandibular3;
public javax.swing.JComboBox jcbSubmandibular4;
public static javax.swing.JLabel jlAlumno;
public static javax.swing.JLabel jlAlumno1;
public static javax.swing.JLabel jlAlumno2;
public static javax.swing.JLabel jlApellido;
public javax.swing.JLabel jlCasoClinico;
public static javax.swing.JLabel jlColor;
public javax.swing.JLabel jlDato;
public static javax.swing.JLabel jlDni;
public static javax.swing.JLabel jlDocente;
public static javax.swing.JLabel jlDocente1;
public static javax.swing.JLabel jlDocente2;
public static javax.swing.JLabel jlDomicilio;
public static javax.swing.JLabel jlEspecie;
public javax.swing.JLabel jlFecha;
public static javax.swing.JLabel jlFechaNacimiento;
public javax.swing.JLabel jlHora;
public static javax.swing.JLabel jlNombre;
public static javax.swing.JLabel jlPeso;
public static javax.swing.JLabel jlRaza;
public static javax.swing.JLabel jlSexo;
public static javax.swing.JLabel jlTelefono;
public javax.swing.JPanel jpAxilar;
public javax.swing.JPanel jpConsulta10;
```



```
public javax.swing.JPanel jpConvulsion;  
public javax.swing.JPanel jpDeshidratado;  
public static javax.swing.JPanel jpGeneral;  
public javax.swing.JPanel jpInguinales;  
public javax.swing.JPanel jpParalisis;  
public javax.swing.JPanel jpParesia;  
public javax.swing.JPanel jpPopliteos;  
public javax.swing.JPanel jpRetorno1;  
public javax.swing.JPanel jpRetorno2;  
public javax.swing.JPanel jpRetorno3;  
public javax.swing.JPanel jpSubmandibular;  
public javax.swing.JTextArea jtaAnamnesis;  
public javax.swing.JTextArea jtaIndicaciones;  
public javax.swing.JTextArea jtaSistema;  
public javax.swing.JTextField jtfDeshidratado;  
public javax.swing.JTextField jtfFC;  
public javax.swing.JTextField jtfFR;  
public javax.swing.JTextField jtfTemperatura;  
public javax.swing.JTabbedPane jtpPestañas;  
// End of variables declaration  
}
```

RetornoAnimalG.Java

```

package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import java.sql.Date;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
public class RetornoAnimalG extends javax.swing.JInternalFrame {
    public RetornoAnimalG() {
        initComponents();
        Calendar Cal= Calendar.getInstance();
        String cc =BuscarCasoClinicoG.auxenviarg.getText();
        String fecha =BuscarCasoClinicoG.auxenviarg2.getText();
        String fecha1=fecha+"0101";
        String fecha2=fecha+"1231";
        jlCasoClinico.setText(""+cc+"/"+Cal.get(Cal.YEAR));
        int cod_animal=Integer.parseInt(animal.GetData("cod_animal","SELECT cod_animal FROM consultag WHERE caso_clinico = \"+cc+"\""));
        jlNombre.setText(animal.GetData("nombre","SELECT nombre FROM animal_grandes WHERE cod_animal = \"+cod_animal+"\""));
        int especie=Integer.parseInt(animal.GetData("cod_especie","SELECT cod_especie FROM animal_grandes WHERE cod_animal = \"+cod_animal+"\""));
        jlEspecie.setText(animal.GetData("especie","SELECT especie FROM lista_especies WHERE cod_especie = \"+especie+"\""));
        int raza=Integer.parseInt(animal.GetData("cod_raza","SELECT cod_raza FROM animal_grandes WHERE cod_animal = \"+cod_animal+"\""));
        jlRaza.setText(animal.GetData("raza","SELECT raza FROM lista_razas WHERE cod_raza = \"+raza+"\""));
        jlPeso.setText(animal.GetData("peso","SELECT peso FROM animal_grandes WHERE cod_animal = \"+cod_animal+"\""));
        jlFechaNacimiento.setText(animal.GetData("nacimiento","SELECT nacimiento FROM animal_grandes WHERE cod_animal = \"+cod_animal+"\""));
        jlSexo.setText(animal.GetData("sexo","SELECT sexo FROM animal_grandes WHERE cod_animal = \"+cod_animal+"\""));
        jlColor.setText(animal.GetData("color","SELECT color FROM animal_grandes WHERE cod_animal = \"+cod_animal+"\""));
        int dueño=Integer.parseInt(animal.GetData("dni_duenio","SELECT dni_duenio FROM animal_grandes WHERE cod_animal = \"+cod_animal+"\""));
        jlDni.setText(animal.GetData("dni_duenio","SELECT dni_duenio FROM animal_grandes WHERE cod_animal = \"+cod_animal+"\""));
        jlApellido.setText(animal.GetData("apellido","SELECT apellido FROM duenio WHERE dni_duenio = \"+dueño+"\""));
        jlDomicilio.setText(animal.GetData("domicilio","SELECT domicilio FROM duenio WHERE dni_duenio = \"+dueño+"\""));
        jlTelefono.setText(animal.GetData("telefono_fijo","SELECT telefono_fijo FROM duenio WHERE dni_duenio = \"+dueño+"\""));
        jlDato.setText(animal.GetData("email","SELECT email FROM duenio WHERE dni_duenio = \"+dueño+"\""));
        String fec= Cal.get(Cal.DATE)+"/"+(Cal.get(Cal.MONTH)+1)+"/"+Cal.get(Cal.YEAR);
        String hor=Cal.get(Cal.HOUR_OF_DAY)+":"+Cal.get(Cal.MINUTE)+":"+Cal.get(Cal.SECOND);
        jlFecha.setText(fec);
        jlHora.setText(hor);
    }
    /*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")
    private void jcbHidratacionActionPerformed(java.awt.event.ActionEvent evt) {
        String hidratado="Deshidratado";
        String hidratado2=(String)jcbHidratacion.getSelectedItem();
        if (hidratado2==hidratado){
            jpDeshidratado.setVisible(true);
        }else{
            jpDeshidratado.setVisible(false);
        }
    }
}

```

```

}
private void jcbSubmandibular1ActionPerformed(java.awt.event.ActionEvent evt) {
    String submandibular = (String)jcbSubmandibular1.getSelectedItemAt();
    String submandibular2 = "No";
    if(submandibular==submandibular2){
        jpSubmandibular.setVisible(true);
    }else{
        jpSubmandibular.setVisible(false);
    }
}
private void jcbAxilar1ActionPerformed(java.awt.event.ActionEvent evt) {
    String axilar = (String)jcbAxilar1.getSelectedItemAt();
    String axilar2 = "No";
    if(axilar==axilar2){
        jpAxilar.setVisible(true);
    }else{
        jpAxilar.setVisible(false);
    }
}
private void jcbInguinales1ActionPerformed(java.awt.event.ActionEvent evt) {
    String inguinales = (String)jcbInguinales1.getSelectedItemAt();
    String inguinales2 = "No";
    if(inguinales==inguinales2){
        jpInguinales.setVisible(true);
    }else{
        jpInguinales.setVisible(false);
    }
}
private void jcbPopliteos1ActionPerformed(java.awt.event.ActionEvent evt) {
    String popliteos = (String)jcbPopliteos1.getSelectedItemAt();
    String popliteos2 = "No";
    if(popliteos==popliteos2){
        jpPopliteos.setVisible(true);
    }else{
        jpPopliteos.setVisible(false);
    }
}
private void jbGuardarActionPerformed(java.awt.event.ActionEvent evt) {
    String caso_clinico = BuscarCasoClinicoG.auxenviarg.getText();
    /* Buscamos el ultimo cod_retorno y se suma 1 es decir, si el ultimo es 2 , se guarda 3
    int Max= Integer.parseInt(animal.GetData("cod_retorno", "SELECT count(cod_retorno)AS cod_retorno FROM retornog"));
    String cod_retorno= Integer.toString(Max+1);
    String cod_animal="1";
    String dni_doctor=jIDocente2.getText();
    String dni_alumno=jIAlumno2.getText();
    boolean aux = animal.existe("retornog", "cod_retorno", cod_retorno);
    if (aux==true){
        JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
    }else{
        String anamnesis = jtaAnamnesis.getText();
        String temperatura = jtftemperatura.getText();
        String fc = jtfFC.getText();
        String fr = jtfFR.getText();
        String cc = (String)jcbCC.getSelectedItemAt();
        String ll_cap = (String)jcbLLCap.getSelectedItemAt();
        String pulso_ritmo = (String)jcbPulsoRitmo.getSelectedItemAt();
        String pulso_intensidad = (String)jcbPulsoIntensidad.getSelectedItemAt();
        String hidratacion2="Deshidratado";
        String hidratacion=(String)jcbHidratacion.getSelectedItemAt();
        if (hidratacion2==hidratacion){
            hidratacion = jtfDeshidratado.getText();
        }else{

```

```

        hidratacion=(String)jcbHidratacion.getSelectedItemAt();
    }
    String ma_ocular = (String)jcbMaOcular.getSelectedItemAt();
    String ma_bucal = (String)jcbMaBucal.getSelectedItemAt();
    String ma_nasal = (String)jcbMaNasal.getSelectedItemAt();
    String ma_genital = (String)jcbMaGenital.getSelectedItemAt();
    String submandibular = (String)jcbSubmandibular1.getSelectedItemAt();
    String submandibular2 = "No";
    if(submandibular==submandibular2){
        submandibular =
        (String)jcbSubmandibular2.getSelectedItemAt()+" "+jcbSubmandibular3.getSelectedItemAt()+" "+jcbSubmandibular4.getSelectedItemAt();
    }else{
        submandibular = "Si";
    }
    String axilar = (String)jcbAxilar1.getSelectedItemAt();
    String axilar2 = "No";
    if(axilar==axilar2){
        axilar = (String)jcbAxilar2.getSelectedItemAt()+" "+jcbAxilar3.getSelectedItemAt()+" "+jcbAxilar4.getSelectedItemAt();
    }else{
        axilar = "Si";
    }
    String inguinales = (String)jcbInguinales1.getSelectedItemAt();
    String inguinales2 = "No";
    if(inguinales==inguinales2){
        inguinales = (String)jcbInguinales2.getSelectedItemAt()+" "+jcbInguinales3.getSelectedItemAt()+" "+jcbInguinales4.getSelectedItemAt();
    }else{
        inguinales = "Si";
    }
    String popliteos = (String)jcbPopliteos1.getSelectedItemAt();
    String popliteos2 = "No";
    if(popliteos==popliteos2){
        popliteos = (String)jcbPopliteos2.getSelectedItemAt()+" "+jcbPopliteos3.getSelectedItemAt()+" "+jcbPopliteos4.getSelectedItemAt();
    }else{
        popliteos = "Si";
    }
    String sistema = jtaSistema.getText();
    String indicaciones = jtaIndicaciones.getText();
    java.util.Date date = new java.util.Date();
    java.text.SimpleDateFormat sdf=new java.text.SimpleDateFormat("yyyy-MM-dd");
    String fecha = sdf.format(date);
    String datos[] =
    {cod_retorno,caso_clinico,fecha,cod_animal,dni_doctor,dni_alumno,anamnesis,temperatura,fc,fr,cc,Il_cap,pulso_ritmo,pulso_intensidad,hidratacion,
    ma_ocular,ma_bucal,ma_nasal,ma_genital,submandibular,axilar,inguinales,popliteos,sistema,indicaciones};
    animal.InsertarRegistro(datos, "insert into retornog values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");
    }
}

private void jBuscarAlumnoActionPerformed(java.awt.event.ActionEvent evt) {
    BuscarAlumnoRetornoG buscarAlumno = new BuscarAlumnoRetornoG();
    PantallaPrincipal.jdpFondo.add(buscarAlumno);
    buscarAlumnoToFront();
    buscarAlumno.setVisible(true);
}

private void jBuscarDocenteActionPerformed(java.awt.event.ActionEvent evt) {
    BuscarDocenteRetornoG buscarDocente = new BuscarDocenteRetornoG();
    PantallaPrincipal.jdpFondo.add(buscarDocente);
    buscarDocenteToFront();
    buscarDocente.setVisible(true);
}

private void jcbDatosDueñoActionPerformed(java.awt.event.ActionEvent evt) {
    String dni= jIdni.getText();
    String dato = jcbDatosDueño.getSelectedItemAt().toString();
    if(dato=="e-mail"){

```

```
jldato.setText(animal.GetData("email","SELECT email FROM duenio WHERE dni_duenio = \"'+dni+'\""));
}
if(dato=="facebook"){
    jldato.setText(animal.GetData("facebook","SELECT facebook FROM duenio WHERE dni_duenio = \"'+dni+'\""));
}
if(dato=="celular"){
    jldato.setText(animal.GetData("telefono_celular","SELECT telefono_celular FROM duenio WHERE dni_duenio = \"'+dni+'\""));
}
}
private void jbSalirActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}
MySQL animal = new MySQL();
// Variables declaration - do not modify
public javax.swing.JButton jButton1;
public javax.swing.JButton jButton2;
public javax.swing.JLabel jLabel1;
public javax.swing.JLabel jLabel10;
public javax.swing.JLabel jLabel11;
public javax.swing.JLabel jLabel116;
public javax.swing.JLabel jLabel12;
public javax.swing.JLabel jLabel123;
public javax.swing.JLabel jLabel124;
public javax.swing.JLabel jLabel125;
public javax.swing.JLabel jLabel126;
public javax.swing.JLabel jLabel13;
public javax.swing.JLabel jLabel14;
public javax.swing.JLabel jLabel143;
public javax.swing.JLabel jLabel15;
public static javax.swing.JLabel jLabel16;
public javax.swing.JLabel jLabel18;
public javax.swing.JLabel jLabel19;
public javax.swing.JLabel jLabel2;
public javax.swing.JLabel jLabel20;
public javax.swing.JLabel jLabel21;
public javax.swing.JLabel jLabel22;
public javax.swing.JLabel jLabel23;
public javax.swing.JLabel jLabel24;
public javax.swing.JLabel jLabel25;
public javax.swing.JLabel jLabel26;
public javax.swing.JLabel jLabel27;
public javax.swing.JLabel jLabel28;
public javax.swing.JLabel jLabel29;
public javax.swing.JLabel jLabel3;
public javax.swing.JLabel jLabel30;
public javax.swing.JLabel jLabel31;
public javax.swing.JLabel jLabel32;
public javax.swing.JLabel jLabel33;
public javax.swing.JLabel jLabel34;
public javax.swing.JLabel jLabel35;
public javax.swing.JLabel jLabel36;
public javax.swing.JLabel jLabel37;
public javax.swing.JLabel jLabel38;
public javax.swing.JLabel jLabel39;
public javax.swing.JLabel jLabel4;
public javax.swing.JLabel jLabel40;
public javax.swing.JLabel jLabel41;
public javax.swing.JLabel jLabel42;
public javax.swing.JLabel jLabel43;
public javax.swing.JLabel jLabel44;
public javax.swing.JLabel jLabel45;
public javax.swing.JLabel jLabel46;
```



```
public javax.swing.JLabel jLabel47;
public javax.swing.JLabel jLabel5;
public javax.swing.JLabel jLabel6;
public javax.swing.JLabel jLabel7;
public javax.swing.JLabel jLabel8;
public javax.swing.JLabel jLabel9;
public javax.swing.JScrollPane jScrollPane1;
public javax.swing.JScrollPane jScrollPane2;
public javax.swing.JScrollPane jScrollPane35;
public static javax.swing.JScrollPane jScrollPane37;
public static javax.swing.JTabbedPane jTabbedPane1;
public javax.swing.JButton jbGuardar;
public javax.swing.JButton jbSalir;
public javax.swing.JComboBox jcbAxilar1;
public javax.swing.JComboBox jcbAxilar2;
public javax.swing.JComboBox jcbAxilar3;
public javax.swing.JComboBox jcbAxilar4;
public javax.swing.JComboBox jcbCC;
public javax.swing.JComboBox jcbDatosDueño;
public javax.swing.JComboBox jcbHidratacion;
public javax.swing.JComboBox jcbInguinales1;
public javax.swing.JComboBox jcbInguinales2;
public javax.swing.JComboBox jcbInguinales3;
public javax.swing.JComboBox jcbInguinales4;
public javax.swing.JComboBox jcbLLCap;
public javax.swing.JComboBox jcbMaBucal;
public javax.swing.JComboBox jcbMaGenital;
public javax.swing.JComboBox jcbMaNasal;
public javax.swing.JComboBox jcbMaOcular;
public javax.swing.JComboBox jcbPopliteos1;
public javax.swing.JComboBox jcbPopliteos2;
public javax.swing.JComboBox jcbPopliteos3;
public javax.swing.JComboBox jcbPopliteos4;
public javax.swing.JComboBox jcbPulsoIntensidad;
public javax.swing.JComboBox jcbPulsoRitmo;
public javax.swing.JComboBox jcbSubmandibular1;
public javax.swing.JComboBox jcbSubmandibular2;
public javax.swing.JComboBox jcbSubmandibular3;
public javax.swing.JComboBox jcbSubmandibular4;
public static javax.swing.JLabel jlAlumno;
public static javax.swing.JLabel jlAlumno1;
public static javax.swing.JLabel jlAlumno2;
public static javax.swing.JLabel jlApellido;
public javax.swing.JLabel jlCasoClinico;
public static javax.swing.JLabel jlColor;
public javax.swing.JLabel jlDato;
public static javax.swing.JLabel jlDni;
public static javax.swing.JLabel jlDocente;
public static javax.swing.JLabel jlDocente1;
public static javax.swing.JLabel jlDocente2;
public static javax.swing.JLabel jlDomicilio;
public static javax.swing.JLabel jlEspecie;
public javax.swing.JLabel jlFecha;
public static javax.swing.JLabel jlFechaNacimiento;
public javax.swing.JLabel jlHora;
public static javax.swing.JLabel jlNombre;
public static javax.swing.JLabel jlPeso;
public static javax.swing.JLabel jlRaza;
public static javax.swing.JLabel jlSexo;
public static javax.swing.JLabel jlTelefono;
public javax.swing.JPanel jpAxilar;
public javax.swing.JPanel jpConsulta10;
```




```
public javax.swing.JPanel jpConvulsion;  
public javax.swing.JPanel jpDeshidratado;  
public static javax.swing.JPanel jpGeneral;  
public javax.swing.JPanel jpInguinales;  
public javax.swing.JPanel jpParalisis;  
public javax.swing.JPanel jpParesia;  
public javax.swing.JPanel jpPopliteos;  
public javax.swing.JPanel jpRetorno1;  
public javax.swing.JPanel jpRetorno2;  
public javax.swing.JPanel jpRetorno3;  
public javax.swing.JPanel jpSubmandibular;  
public javax.swing.JTextArea jtaAnamnesis;  
public javax.swing.JTextArea jtaIndicaciones;  
public javax.swing.JTextArea jtaSistema;  
public javax.swing.JTextField jtfDeshidratado;  
public javax.swing.JTextField jtfFC;  
public javax.swing.JTextField jtfFR;  
public javax.swing.JTextField jtftemperatura;  
public javax.swing.JTabbedPane jtpPestañas;  
// End of variables declaration  
}
```

RetornoAnimalP.Java

```

package Consultas;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import java.sql.Date;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
public class RetornoAnimalP extends javax.swing.JInternalFrame {
    public RetornoAnimalP() {
        initComponents();
        Calendar Cal= Calendar.getInstance();
        String cc =BuscarCasoClinico.auxenviar.getText();
        String fecha =BuscarCasoClinico.auxenviar1.getText();
        String fecha1=fecha+"0101";
        String fecha2=fecha+"1231";
        jlCasoClinico.setText(""+BuscarCasoClinico.auxenviar.getText()+"/"+Cal.get(Cal.YEAR));
        int cod_animal=Integer.parseInt(animal.GetData("cod_animal","SELECT cod_animal FROM consulta WHERE caso_clinico = \''"+cc+"\'"));
        jlNombre.setText(animal.GetData("nombre","SELECT nombre FROM animal_pequenos WHERE cod_animal = \''"+cod_animal+"\'"));
        int especie=Integer.parseInt(animal.GetData("cod_especie","SELECT cod_especie FROM animal_pequenos WHERE cod_animal =
\'"+cod_animal+"\'"));
        jlEspecie.setText(animal.GetData("especie","SELECT especie FROM lista_especie WHERE cod_especie = \''"+especie+"\'"));
        int raza=Integer.parseInt(animal.GetData("cod_raza","SELECT cod_raza FROM animal_pequenos WHERE cod_animal = \''"+cod_animal+"\'"));
        jlRaza.setText(animal.GetData("raza","SELECT raza FROM lista_raza WHERE cod_raza = \''"+raza+"\'"));
        jlPeso.setText(animal.GetData("peso","SELECT peso FROM animal_pequenos WHERE cod_animal = \''"+cod_animal+"\'"));
        jlFechaNacimiento.setText(animal.GetData("nacimiento","SELECT nacimiento FROM animal_pequenos WHERE cod_animal =
\'"+cod_animal+"\'"));
        jlSexo.setText(animal.GetData("sexo","SELECT sexo FROM animal_pequenos WHERE cod_animal = \''"+cod_animal+"\'"));
        jlColor.setText(animal.GetData("color","SELECT color FROM animal_pequenos WHERE cod_animal = \''"+cod_animal+"\'"));
        int duenio=Integer.parseInt(animal.GetData("dni_duenio","SELECT dni_duenio FROM animal_pequenos WHERE cod_animal =
\'"+cod_animal+"\'"));
        jlDni.setText(animal.GetData("dni_duenio","SELECT dni_duenio FROM animal_pequenos WHERE cod_animal = \''"+cod_animal+"\'"));
        jlApellido.setText(animal.GetData("apellido","SELECT apellido FROM duenio WHERE dni_duenio = \''"+duenio+"\'"));
        jlDomicilio.setText(animal.GetData("domicilio","SELECT domicilio FROM duenio WHERE dni_duenio = \''"+duenio+"\'"));
        jlTelefono.setText(animal.GetData("telefono_fijo","SELECT telefono_fijo FROM duenio WHERE dni_duenio = \''"+duenio+"\'"));
        jlDato.setText(animal.GetData("email","SELECT email FROM duenio WHERE dni_duenio = \''"+duenio+"\'"));
        String fec= Cal.get(Cal.DATE)+"/"+(Cal.get(Cal.MONTH)+1)+"/"+Cal.get(Cal.YEAR);
        String hor=Cal.get(Cal.HOUR_OF_DAY)+":"+Cal.get(Cal.MINUTE)+":"+Cal.get(Cal.SECOND);
        jlFecha.setText(fec);
        jlHora.setText(hor);
    }
    /*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")
    private void jcbHidratacionActionPerformed(java.awt.event.ActionEvent evt) {
        String hidratado="Deshidratado";
        String hidratado2=(String)jcbHidratacion.getSelectedItem();
        if (hidratado2==hidratado){
            jpDeshidratado.setVisible(true);
        }else{
            jpDeshidratado.setVisible(false);
        }
    }
}

```

```

}
private void jcbSubmandibular1ActionPerformed(java.awt.event.ActionEvent evt) {
    String submandibular = (String)jcbSubmandibular1.getSelectedItemAt();
    String submandibular2 = "No";
    if(submandibular==submandibular2){
        jpSubmandibular.setVisible(true);
    }else{
        jpSubmandibular.setVisible(false);
    }
}
private void jcbAxilar1ActionPerformed(java.awt.event.ActionEvent evt) {
    String axilar = (String)jcbAxilar1.getSelectedItemAt();
    String axilar2 = "No";
    if(axilar==axilar2){
        jpAxilar.setVisible(true);
    }else{
        jpAxilar.setVisible(false);
    }
}
private void jcbInguinales1ActionPerformed(java.awt.event.ActionEvent evt) {
    String inguinales = (String)jcbInguinales1.getSelectedItemAt();
    String inguinales2 = "No";
    if(inguinales==inguinales2){
        jpInguinales.setVisible(true);
    }else{
        jpInguinales.setVisible(false);
    }
}
private void jcbPopliteos1ActionPerformed(java.awt.event.ActionEvent evt) {
    String popliteos = (String)jcbPopliteos1.getSelectedItemAt();
    String popliteos2 = "No";
    if(popliteos==popliteos2){
        jpPopliteos.setVisible(true);
    }else{
        jpPopliteos.setVisible(false);
    }
}
private void jbGuardarActionPerformed(java.awt.event.ActionEvent evt) {
    String caso_clinico = BuscarCasoClinico.auxenviar.getText();
    /* Buscamos el ultimo cod_retorno y se suma 1 es decir, si el ultimo es 2 , se guarda 3
    int Max= Integer.parseInt(animal.GetData("cod_retorno", "SELECT count(cod_retorno)AS cod_retorno FROM retorno"));
    String cod_retorno= Integer.toString(Max+1);
    String cod_animal="1";
    String dni_doctor=jlDocente2.getText();
    String dni_alumno=jlAlumno2.getText();
    boolean aux = animal.existe("retorno", "cod_retorno", cod_retorno);
    if (aux==true){
        JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
    }else{
        String anamnesis = jtaAnamnesis.getText();
        String temperatura = jtftemperatura.getText();
        String fc = jtfFC.getText();
        String fr = jtfFR.getText();
        String cc = (String)jcbCC.getSelectedItemAt();
        String ll_cap = (String)jcbLLCap.getSelectedItemAt();
        String pulso_ritmo = (String)jcbPulsoRitmo.getSelectedItemAt();
        String pulso_intensidad = (String)jcbPulsoIntensidad.getSelectedItemAt();
        String hidratacion2="Deshidratado";
        String hidratacion=(String)jcbHidratacion.getSelectedItemAt();
        if (hidratacion2==hidratacion){
            hidratacion = jtfDeshidratado.getText();
        }else{

```

```

        hidratacion=(String)jcbHidratacion.getSelectedItemAt();
    }
    String ma_ocular = (String)jcbMaOcular.getSelectedItemAt();
    String ma_bucal = (String)jcbMaBucal.getSelectedItemAt();
    String ma_nasal = (String)jcbMaNasal.getSelectedItemAt();
    String ma_genital = (String)jcbMaGenital.getSelectedItemAt();
    String submandibular = (String)jcbSubmandibular1.getSelectedItemAt();
    String submandibular2 = "No";
    if(submandibular==submandibular2){
        submandibular =
        (String)jcbSubmandibular2.getSelectedItemAt()+" "+jcbSubmandibular3.getSelectedItemAt()+" "+jcbSubmandibular4.getSelectedItemAt();
    }else{
        submandibular = "Si";
    }
    String axilar = (String)jcbAxilar1.getSelectedItemAt();
    String axilar2 = "No";
    if(axilar==axilar2){
        axilar = (String)jcbAxilar2.getSelectedItemAt()+" "+jcbAxilar3.getSelectedItemAt()+" "+jcbAxilar4.getSelectedItemAt();
    }else{
        axilar = "Si";
    }
    String inguinales = (String)jcbInguinales1.getSelectedItemAt();
    String inguinales2 = "No";
    if(inguinales==inguinales2){
        inguinales = (String)jcbInguinales2.getSelectedItemAt()+" "+jcbInguinales3.getSelectedItemAt()+" "+jcbInguinales4.getSelectedItemAt();
    }else{
        inguinales = "Si";
    }
    String popliteos = (String)jcbPopliteos1.getSelectedItemAt();
    String popliteos2 = "No";
    if(popliteos==popliteos2){
        popliteos = (String)jcbPopliteos2.getSelectedItemAt()+" "+jcbPopliteos3.getSelectedItemAt()+" "+jcbPopliteos4.getSelectedItemAt();
    }else{
        popliteos = "Si";
    }
    String sistema = jtaSistema.getText();
    String indicaciones = jtaIndicaciones.getText();
    java.util.Date date = new java.util.Date();
    java.text.SimpleDateFormat sdf=new java.text.SimpleDateFormat("yyyy-MM-dd");
    String fecha = sdf.format(date);
    String datos[] =
    {cod_retorno,caso_clinico,fecha,cod_animal,dni_doctor,dni_alumno,anamnesis,temperatura,fc,fr,cc,ll_cap,pulso_ritmo,pulso_intensidad,hidratacion,
    ma_ocular,ma_bucal,ma_nasal,ma_genital,submandibular,axilar,inguinales,popliteos,sistema,indicaciones};
    animal.InsertarRegistro(datos, "insert into retorno values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");
    }
}

private void jBuscarAlumnoActionPerformed(java.awt.event.ActionEvent evt) {
    BuscarAlumnoRetorno buscarAlumno = new BuscarAlumnoRetorno();
    PantallaPrincipal.jdpFondo.add(buscarAlumno);
    buscarAlumnoToFront();
    buscarAlumno.setVisible(true);
}

private void jBuscarDocenteActionPerformed(java.awt.event.ActionEvent evt) {
    BuscarDocenteRetorno buscarDocente = new BuscarDocenteRetorno();
    PantallaPrincipal.jdpFondo.add(buscarDocente);
    buscarDocenteToFront();
    buscarDocente.setVisible(true);
}

private void jcbDatosDueñoActionPerformed(java.awt.event.ActionEvent evt) {
    String dni= jIdni.getText();
    String dato = jcbDatosDueño.getSelectedItemAt().toString();
    if(dato=="e-mail"){

```

```
jIDato.setText(animal.GetData("email","SELECT email FROM duenio WHERE dni_duenio = \"'+dni+'\""));
}
if(dato=="facebook"){
    jIDato.setText(animal.GetData("facebook","SELECT facebook FROM duenio WHERE dni_duenio = \"'+dni+'\""));
}
if(dato=="celular"){
    jIDato.setText(animal.GetData("telefono_celular","SELECT telefono_celular FROM duenio WHERE dni_duenio = \"'+dni+'\""));
}
}
private void jbSalirActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}
MySQL animal = new MySQL();
// Variables declaration - do not modify
public javax.swing.JButton jButton1;
public javax.swing.JButton jButton2;
public javax.swing.JLabel jLabel1;
public javax.swing.JLabel jLabel10;
public javax.swing.JLabel jLabel11;
public javax.swing.JLabel jLabel116;
public javax.swing.JLabel jLabel12;
public javax.swing.JLabel jLabel123;
public javax.swing.JLabel jLabel124;
public javax.swing.JLabel jLabel125;
public javax.swing.JLabel jLabel126;
public javax.swing.JLabel jLabel13;
public javax.swing.JLabel jLabel14;
public javax.swing.JLabel jLabel143;
public javax.swing.JLabel jLabel15;
public static javax.swing.JLabel jLabel16;
public javax.swing.JLabel jLabel18;
public javax.swing.JLabel jLabel19;
public javax.swing.JLabel jLabel2;
public javax.swing.JLabel jLabel20;
public javax.swing.JLabel jLabel21;
public javax.swing.JLabel jLabel22;
public javax.swing.JLabel jLabel23;
public javax.swing.JLabel jLabel24;
public javax.swing.JLabel jLabel25;
public javax.swing.JLabel jLabel26;
public javax.swing.JLabel jLabel27;
public javax.swing.JLabel jLabel28;
public javax.swing.JLabel jLabel29;
public javax.swing.JLabel jLabel3;
public javax.swing.JLabel jLabel30;
public javax.swing.JLabel jLabel31;
public javax.swing.JLabel jLabel32;
public javax.swing.JLabel jLabel33;
public javax.swing.JLabel jLabel34;
public javax.swing.JLabel jLabel35;
public javax.swing.JLabel jLabel36;
public javax.swing.JLabel jLabel37;
public javax.swing.JLabel jLabel38;
public javax.swing.JLabel jLabel39;
public javax.swing.JLabel jLabel4;
public javax.swing.JLabel jLabel40;
public javax.swing.JLabel jLabel41;
public javax.swing.JLabel jLabel42;
public javax.swing.JLabel jLabel43;
public javax.swing.JLabel jLabel44;
public javax.swing.JLabel jLabel45;
public javax.swing.JLabel jLabel46;
```

```
public javax.swing.JLabel jLabel47;
public javax.swing.JLabel jLabel5;
public javax.swing.JLabel jLabel6;
public javax.swing.JLabel jLabel7;
public javax.swing.JLabel jLabel8;
public javax.swing.JLabel jLabel9;
public javax.swing.JScrollPane jScrollPane1;
public javax.swing.JScrollPane jScrollPane2;
public javax.swing.JScrollPane jScrollPane35;
public static javax.swing.JScrollPane jScrollPane37;
public static javax.swing.JTabbedPane jTabbedPane1;
public javax.swing.JButton jbGuardar;
public javax.swing.JButton jbSalir;
public javax.swing.JComboBox jcbAxilar1;
public javax.swing.JComboBox jcbAxilar2;
public javax.swing.JComboBox jcbAxilar3;
public javax.swing.JComboBox jcbAxilar4;
public javax.swing.JComboBox jcbCC;
public javax.swing.JComboBox jcbDatosDueño;
public javax.swing.JComboBox jcbHidratacion;
public javax.swing.JComboBox jcbInguinales1;
public javax.swing.JComboBox jcbInguinales2;
public javax.swing.JComboBox jcbInguinales3;
public javax.swing.JComboBox jcbInguinales4;
public javax.swing.JComboBox jcbLLCap;
public javax.swing.JComboBox jcbMaBucal;
public javax.swing.JComboBox jcbMaGenital;
public javax.swing.JComboBox jcbMaNasal;
public javax.swing.JComboBox jcbMaOcular;
public javax.swing.JComboBox jcbPopliteos1;
public javax.swing.JComboBox jcbPopliteos2;
public javax.swing.JComboBox jcbPopliteos3;
public javax.swing.JComboBox jcbPopliteos4;
public javax.swing.JComboBox jcbPulsoIntensidad;
public javax.swing.JComboBox jcbPulsoRitmo;
public javax.swing.JComboBox jcbSubmandibular1;
public javax.swing.JComboBox jcbSubmandibular2;
public javax.swing.JComboBox jcbSubmandibular3;
public javax.swing.JComboBox jcbSubmandibular4;
public static javax.swing.JLabel jlAlumno;
public static javax.swing.JLabel jlAlumno1;
public static javax.swing.JLabel jlAlumno2;
public static javax.swing.JLabel jlApellido;
public javax.swing.JLabel jlCasoClinico;
public static javax.swing.JLabel jlColor;
public javax.swing.JLabel jlDato;
public static javax.swing.JLabel jlDni;
public static javax.swing.JLabel jlDocente;
public static javax.swing.JLabel jlDocente1;
public static javax.swing.JLabel jlDocente2;
public static javax.swing.JLabel jlDomicilio;
public static javax.swing.JLabel jlEspecie;
public javax.swing.JLabel jlFecha;
public static javax.swing.JLabel jlFechaNacimiento;
public javax.swing.JLabel jlHora;
public static javax.swing.JLabel jlNombre;
public static javax.swing.JLabel jlPeso;
public static javax.swing.JLabel jlRaza;
public static javax.swing.JLabel jlSexo;
public static javax.swing.JLabel jlTelefono;
public javax.swing.JPanel jpAxilar;
public javax.swing.JPanel jpConsulta10;
```



```
public javax.swing.JPanel jpConvulsion;  
public javax.swing.JPanel jpDeshidratado;  
public static javax.swing.JPanel jpGeneral;  
public javax.swing.JPanel jpInguinales;  
public javax.swing.JPanel jpParalisis;  
public javax.swing.JPanel jpParesia;  
public javax.swing.JPanel jpPopliteos;  
public javax.swing.JPanel jpRetorno1;  
public javax.swing.JPanel jpRetorno2;  
public javax.swing.JPanel jpRetorno3;  
public javax.swing.JPanel jpSubmandibular;  
public javax.swing.JTextArea jtaAnamnesis;  
public javax.swing.JTextArea jtaIndicaciones;  
public javax.swing.JTextArea jtaSistema;  
public javax.swing.JTextField jtfDeshidratado;  
public javax.swing.JTextField jtfFC;  
public javax.swing.JTextField jtfFR;  
public javax.swing.JTextField jtftemperatura;  
public javax.swing.JTabbedPane jtpPestañas;  
// End of variables declaration  
}
```


Paquetes Exámenes

BuscarCasoClinico.Java

```

package Examenes;
import Examenes.*;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import ModuloUsuario.ImgDoctor;
import javax.swing.table.DefaultTableModel;
public class BuscarCasoClinico extends javax.swing.JInternalFrame {
    public BuscarCasoClinico() {
        initComponents();
        ImgDoctor img = new ImgDoctor();
        jplImagen.add(img);
        jplImagen.repaint();
        cargarTabla();
        jlBuscarCasoClinico.setVisible(false);
    }
    private void cargarTabla(){
        String[] columnNames = {"Caso Clinico", "Animal", "Nombre del Animal", "DNI del Dueño", "Apellido del Dueño", "Nombre del Dueño", "DNI del
        Doctor", "Apellido del Doctor", "Nombre del Doctor"};
        String colName[] = {"caso_clinico", "cod_animal", "animal_pequenios.nombre", "dni_duenio", "duenio.apellido", "duenio.nombre",
        "dni_doctor", "doctor.apellido", "doctor.nombre"};
        //nombre de columnas, tabla, instruccion sql
        dtCasoClinico = casoClinico.GetTabla(colName, "consulta,duenio,animal_pequenios,doctor", "SELECT * FROM
        `consulta`,`duenio`,`animal_pequenios`,`doctor` WHERE (`duenio`.`dni_duenio` = `animal_pequenios`.`dni_duenio`) AND (`consulta`.`dni_doctor`
        = `doctor`.`dni_doctor`) AND (`consulta`.`cod_animal` = `animal_pequenios`.`cod_animal`)ORDER BY `caso_clinico`");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtCasoClinico, columnNames);
        jtCasoClinico.setModel(datos);
        // se colocan los datos en la tabla
        int a = jtCasoClinico.getRowCount();
        //depurador de jtDuenio de item null
        int i=0;
        a--;
        while(a>=i){
            if(jtCasoClinico.getValueAt(i,1)==null){
                ((DefaultTableModel)jtCasoClinico.getModel()).removeRow(i);
                a--;
                i--;
            }
            i++;
        }
    }
}
/*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")
private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
    int filamodificar= jtCasoClinico.getSelectedRow();
    if(filamodificar>=0){
        String casoClinico= jtCasoClinico.getValueAt(filamodificar, 0).toString();
        String cod_animal= jtCasoClinico.getValueAt(filamodificar, 1).toString();
        String animal_nombre= jtCasoClinico.getValueAt(filamodificar, 2).toString();
        String dni_duenio= jtCasoClinico.getValueAt(filamodificar, 3).toString();
        String duenio_apellido= jtCasoClinico.getValueAt(filamodificar, 4).toString();
        String duenio_nombre= jtCasoClinico.getValueAt(filamodificar, 5).toString();
        String buscarcasoclinico =(String)jlBuscarCasoClinico.getText();
        if(buscarcasoclinico=="1"){
            Hemograma.jlFichaClinica.setText(casoClinico);
        }
    }
}

```

```

Hemograma.jlIDAnimal.setText(animal_nombre);
Hemograma.jlCodAnimal.setText(cod_animal);
Hemograma.jlPropietario.setText(duenio_apellido+" "+duenio_nombre);
Hemograma.jlDniDueño.setText(dni_duenio);
}
if(buscarcasoclinico=="2"){
QuimicaClinica.jlFichaClinica.setText(casoClinico);
QuimicaClinica.jlIDAnimal.setText(animal_nombre);
QuimicaClinica.jlCodAnimal.setText(cod_animal);
QuimicaClinica.jlPropietario.setText(duenio_apellido+" "+duenio_nombre);
QuimicaClinica.jlDniDueño.setText(dni_duenio);
}
if(buscarcasoclinico=="3"){
Urianalis.jlFichaClinica.setText(casoClinico);
Urianalis.jlIDAnimal.setText(animal_nombre);
Urianalis.jlCodAnimal.setText(cod_animal);
Urianalis.jlPropietario.setText(duenio_apellido+" "+duenio_nombre);
Urianalis.jlDniDueño.setText(dni_duenio);
}
}
this.dispose();
}
private void jtCasoClinicoMouseClicked(java.awt.event.MouseEvent evt) {
fila=jtCasoClinico.rowAtPoint(evt.getPoint());
}
private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
this.dispose();
}
private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
String texto= jtfBuscar.getText();
String[] columNames = {"Caso Clinico", "Animal", "Nombre del Animal", "DNI del Dueño", "Apellido del Dueño", "Nombre del Dueño", "DNI del
Doctor", "Apellido del Doctor", "Nombre del Doctor"};
String colName[] = {"caso_clinico", "cod_animal", "animal_pequenos.nombre", "dni_duenio", "duenio.apellido", "duenio.nombre",
"dni_doctor", "doctor.apellido", "doctor.nombre"};
//nombre de columnas, tabla, instruccion sql
if(jrbDni.isSelected()==true){
dtCasoClinico = casoClinico.GetTabla(colName, "consulta,duenio,animal_pequenos,doctor", "SELECT * FROM
`consulta`,`duenio`,`animal_pequenos`,`doctor` WHERE (`duenio`.`dni_duenio`=`animal_pequenos`.`dni_duenio`) AND (`consulta`.`dni_doctor`
=`doctor`.`dni_doctor`) AND (`consulta`.`dni_duenio`=`duenio`.`dni_duenio`)AND (`consulta`.`cod_animal`=`animal_pequenos`.`cod_animal`) AND
`consulta`.`dni_doctor`='"+texto+" ";
}
if(jrbApellido.isSelected()==true){
dtCasoClinico = casoClinico.GetTabla(colName, "consulta,duenio,animal_pequenos,doctor", "SELECT * FROM
`consulta`,`duenio`,`animal_pequenos`,`doctor` WHERE (`duenio`.`dni_duenio`=`animal_pequenos`.`dni_duenio`) AND (`consulta`.`dni_doctor`
=`doctor`.`dni_doctor`) AND (`consulta`.`dni_duenio`=`duenio`.`dni_duenio`)AND (`consulta`.`cod_animal`=`animal_pequenos`.`cod_animal`) AND
`caso_clinico`='"+texto+" ";
}
DefaultTableModel datos = new DefaultTableModel(dtCasoClinico,columNames);
jtCasoClinico.setModel(datos);// se colocan los datos en la tabla
//depurador de jtDuenio de item null
int a = jtCasoClinico.getRowCount();
int i=0;
a--;
while(a>=i){
if(jtCasoClinico.getValueAt(i,1)==null){
((DefaultTableModel)jtCasoClinico.getModel()).removeRow(i);
a--;
i--;
}
i++;
}
}
Object[][] dtCasoClinico;
MySQL casoClinico = new MySQL();
int fila = -1;
// Variables declaration - do not modify
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JButton jbBuscar;

```



```
private javax.swing.JButton jbCancelar;  
private javax.swing.JButton jbElegir;  
public static javax.swing.JLabel jlBuscarCasoClinico;  
private javax.swing.JPanel jlpmagen;  
private javax.swing.JRadioButton jrbApellido;  
private javax.swing.JRadioButton jrbDni;  
public static javax.swing.JTable jtCasoClinico;  
private javax.swing.JTextField jtfBuscar;  
// End of variables declaration  
}
```

BuscarDocente.Java

```
package Examenes;
import Examenes.*;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import ModuloUsuario.ImgDoctor;
import javax.swing.table.DefaultTableModel;
public class BuscarDocente extends javax.swing.JFrame {
    public BuscarDocente() {
        initComponents();
        ImgDoctor img = new ImgDoctor();
        jplImagen.add(img);
        jplImagen.repaint();
        cargarTabla();
        jlBuscarDocente.setVisible(false);
    }
    private void cargarTabla(){
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_doctor", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        dtDoctor = doctor.GetTabla(colName, "doctor", "SELECT * FROM `doctor` ORDER BY `apellido`");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtDoctor, columnNames);
        jtDoctor.setModel(datos);
    }
    /*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")
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar= jtDoctor.getSelectedRow();
        if(filamodificar>=0){
            String apellido= jtDoctor.getValueAt(filamodificar, 0).toString();
            String nombre= jtDoctor.getValueAt(filamodificar, 1).toString();
            String dni= jtDoctor.getValueAt(filamodificar, 2).toString();
            String buscardocente =(String)jlBuscarDocente.getText();
            if(buscardocente=="1"){
                Hemograma.jlResponsable.setText(apellido+" "+nombre);
                Hemograma.jlDniDoctor.setText(dni);
            }
            if(buscardocente=="2"){
                QuimicaClinica.jlResponsable.setText(apellido+" "+nombre);
                QuimicaClinica.jlDniDoctor.setText(dni);
            }
            if(buscardocente=="3"){
                Urianalisis.jlResponsable.setText(apellido+" "+nombre);
                Urianalisis.jlDniDoctor.setText(dni);
            }
        }
        this.dispose();
    }
    private void jtDoctorMouseClicked(java.awt.event.MouseEvent evt) {
        fila=jtDoctor.rowAtPoint(evt.getPoint());
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String texto= jtfBuscar.getText();
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_doctor", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        if(jrbDni.isSelected()==true){
            dtDoctor = doctor.GetTabla(colName, "doctor", "SELECT * FROM `doctor` WHERE `doctor`.`dni_doctor`='"+texto+"'");
        }
        if(jrbApellido.isSelected()==true){
            dtDoctor = doctor.GetTabla(colName, "doctor", "SELECT * FROM `doctor` WHERE `apellido` = '\" + texto + '\"");
        }
    }
}
```

```
}
DefaultTableModel datos = new DefaultTableModel(dtDoctor,columnNames);
jtDoctor.setModel(datos); // se colocan los datos en la tabla
//depurador de jtDuenio de item null
int a = jtDoctor.getRowCount();
int i=0;
a--;
while(a>=i){
    if(jtDoctor.getValueAt(i,1)==null){
        ((DefaultTableModel)jtDoctor.getModel()).removeRow(i);
        a--;
        i--;
    }
    i++;
}
}
Object[][] dtDoctor;
MySQL doctor = new MySQL();
int fila = -1;
// Variables declaration - do not modify
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JButton jbBuscar;
private javax.swing.JButton jbCancelar;
private javax.swing.JButton jbElegir;
public static javax.swing.JLabel jlBuscarDocente;
private javax.swing.JPanel jPanelImagen;
private javax.swing.JRadioButton jrbApellido;
private javax.swing.JRadioButton jrbDni;
public static javax.swing.JTable jtDoctor;
private javax.swing.JTextField jtfBuscar;
// End of variables declaration
}
```

Hemograma.Java

```
package Examenes;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import Consultas.BuscarDueñoP;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
public class Hemograma extends javax.swing.JInternalFrame {
    public Hemograma() {
        initComponents();
        Calendar Cal= Calendar.getInstance();
        String fec= Cal.get(Calendar.DATE)+" / "+(Cal.get(Calendar.MONTH)+1)+" / "+Cal.get(Calendar.YEAR);
        jlFecha.setText(fec);
        jlCodAnimal.setVisible(false);
        jlDniDueño.setVisible(false);
        jlDniDoctor.setVisible(false);
    }
    /*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")
    private void jbBuscar1ActionPerformed(java.awt.event.ActionEvent evt) {
        BuscarCasoClinico buscar = new BuscarCasoClinico();
        PantallaPrincipal.jdpFondo.add(buscar);
        BuscarCasoClinico.jlBuscarCasoClinico.setText("1");
        buscarToFront();
        buscar.setVisible(true);
    }
    private void jbBuscar2ActionPerformed(java.awt.event.ActionEvent evt) {
        BuscarDocente buscarDocente = new BuscarDocente();
        PantallaPrincipal.jdpFondo.add(buscarDocente);
        BuscarDocente.jlBuscarDocente.setText("1");
        buscarDocenteToFront();
        buscarDocente.setVisible(true);
    }
    private void jbRegistrarActionPerformed(java.awt.event.ActionEvent evt) {
        String protocolo_lab = jlProtocolo.getText();
        SimpleDateFormat formatoDelTexto = new SimpleDateFormat("yyyy-MM-dd");
        Calendar Cal= Calendar.getInstance();
        String fecha= formatoDelTexto.format(Cal.getTime());
        String caso_clinico = jlFichaClinica.getText();
        String cod_animal = jlCodAnimal.getText();
        String dni_doctor = jlDniDoctor.getText();
        String eritrocitos= jtfEritrocitos.getText();
        String hemoglobina= jtfHemoglobina.getText();
        String hematocrito= jtfHematocrito.getText();
        String vcm= jtfVCM.getText();
        String hcm= jtfHCM.getText();
        String chcm= jtfCHCM.getText();
        String reticulocitos= jtfReticulocitos.getText();
        String ipr= jtfIPR.getText();
        String eritr_nucleados= jtfEritrNucleados.getText();
        String p_plasmaticas= jtfPPlasmaticas.getText();
        String fibrinogeno= jtfFibrinogeno.getText();
        String rel_pp_fibr= jtfRelPpFibr.getText();
        String leucocitos= jtfLeucocitos.getText();
        String mieloblastos= jtfMieloblastos.getText();
        String promielocitos= jtfPromielocitos.getText();
        String mielocitos= jtfMielocitos.getText();
        String metamielocitos= jtfMetamielocitos.getText();
        String neutrof_cay= jtfNeutrofCay.getText();
        String neutrof_seg= jtfNeutrofSeg.getText();
        String eosinofilos= jtfEosinofilos.getText();
        String basofilos= jtfBasofilos.getText();
    }
}
```

```

String linfocitos= jtfLinfocitos.getText();
String monocitos= jtfMonocitos.getText();
String observaciones= jtaObservaciones.getText();
String datos[] =
{protocolo_lab,fecha,caseo_clinico,cod_animal,dni_doctor,eritrocitos,hemoglobina,hematocrito,vcm,hcm,chcm,reticulocitos,ipr,eritr_nucleados,p_pla
smaticas,fibrinogeno,rel_pp_fibr,leucocitos,mieloblastos,promielocitos,mielocitos,metamielocitos,neutrof_cay,neutrof_seg,eosinofilos,basofilos,linfo
citos,monocitos,observaciones};
    analisis.InsertarRegistro(datos, "insert into hemograma values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");
}
MySQL analisis = new MySQL();
// Variables declaration - do not modify
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel17;
private javax.swing.JLabel jLabel18;
private javax.swing.JLabel jLabel19;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel20;
private javax.swing.JLabel jLabel21;
private javax.swing.JLabel jLabel22;
private javax.swing.JLabel jLabel23;
private javax.swing.JLabel jLabel24;
private javax.swing.JLabel jLabel25;
private javax.swing.JLabel jLabel26;
private javax.swing.JLabel jLabel27;
private javax.swing.JLabel jLabel28;
private javax.swing.JLabel jLabel29;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel30;
private javax.swing.JLabel jLabel31;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
public static javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
public static javax.swing.JTabbedPane jTabbedPane1;
private javax.swing.JButton jbBuscar1;
private javax.swing.JButton jbBuscar2;
private javax.swing.JButton jbRegistrar;
public static javax.swing.JLabel jlAnalisisAnteriores;
public static javax.swing.JLabel jlCodAnimal;
public static javax.swing.JLabel jlDniDoctor;
public static javax.swing.JLabel jlDniDueño;
private javax.swing.JLabel jlFecha;
public static javax.swing.JLabel jlFichaClinica;
public static javax.swing.JLabel jlIDAnimal;
public static javax.swing.JLabel jlIPropietario;
private javax.swing.JLabel jlProtocolo;
public static javax.swing.JLabel jlResponsable;
private javax.swing.JTextArea jtaObservaciones;
private javax.swing.JTextField jtfBasofilos;
private javax.swing.JTextField jtfCHCM;
private javax.swing.JTextField jtfEosinofilos;
private javax.swing.JTextField jtfEritrNucleados;
private javax.swing.JTextField jtfEritrocitos;
private javax.swing.JTextField jtfFibrinogeno;
private javax.swing.JTextField jtfHCM;
private javax.swing.JTextField jtfHematocrito;
private javax.swing.JTextField jtfHemoglobina;
private javax.swing.JTextField jtfIPR;
private javax.swing.JTextField jtfLeucocitos;

```




```
private javax.swing.JTextField jtflInfocitos;  
private javax.swing.JTextField jtfMetamielocitos;  
private javax.swing.JTextField jtfMieloblastos;  
private javax.swing.JTextField jtfMielocitos;  
private javax.swing.JTextField jtfMonocitos;  
private javax.swing.JTextField jtfNeutrofCay;  
private javax.swing.JTextField jtfNeutrofSeg;  
private javax.swing.JTextField jtfPPlasmaticas;  
private javax.swing.JTextField jtfPromielocitos;  
private javax.swing.JTextField jtfRelPpFibr;  
private javax.swing.JTextField jtfReticulocitos;  
private javax.swing.JTextField jtfVCM;  
// End of variables declaration  
}
```

QuimicaClinica.Java

```
package Exámenes;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import java.util.Calendar;
public class QuimicaClinica extends javax.swing.JInternalFrame {
    public QuimicaClinica() {
        initComponents();
        Calendar Cal= Calendar.getInstance();
        String fec= Cal.get(Calendar.DATE)+"/"+(Cal.get(Calendar.MONTH)+1)+"/"+Cal.get(Calendar.YEAR);
        jlFecha.setText(fec);
        jlCodAnimal.setVisible(false);
        jlDniDueño.setVisible(false);
        jlDniDoctor.setVisible(false);
    }
    /*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")
    private void jbBuscar1ActionPerformed(java.awt.event.ActionEvent evt) {
        BuscarCasoClinico buscar = new BuscarCasoClinico();
        PantallaPrincipal.jdpFondo.add(buscar);
        BuscarCasoClinico.jlBuscarCasoClinico.setText("2");
        buscarToFront();
        buscar.setVisible(true);
    }
    private void jbBuscar2ActionPerformed(java.awt.event.ActionEvent evt) {
        BuscarDocente buscarDocente = new BuscarDocente();
        PantallaPrincipal.jdpFondo.add(buscarDocente);
        BuscarDocente.jlBuscarDocente.setText("2");
        buscarDocenteToFront();
        buscarDocente.setVisible(true);
    }
    private void jbRegistrarActionPerformed(java.awt.event.ActionEvent evt) {
        String protocolo_lab = jlProtocolo.getText();
        String fecha = jlFecha.getText();
        String caso_clinico = jlFichaClinica.getText();
        String cod_animal = jlCodAnimal.getText();
        String dni_doctor = jlDniDoctor.getText();
        String glucemia = jtfGlucemia.getText();
        String uremia = jtfUremia.getText();
        String creatininemia = jtfCreatininemia.getText();
        String fosfatemia = jtfFosfatemia.getText();
        String albuminemia = jtfAlbuminemia.getText();
        String got = jtfGOT.getText();
        String gpt = jtfGPT.getText();
        String cpk = jtfCPK.getText();
        String ldh = jtfLDH.getText();
        String observaciones= jtaObservaciones.getText();
        String datos[] =
        {protocolo_lab,fecha,caso_clinico,cod_animal,dni_doctor,glucemia,uremia,creatininemia,fosfatemia,albuminemia,got,gpt,cpk,ldh,observaciones};
        analisis.InsertarRegistro(datos, "insert into quimica_clinica values(?,?,?,?,?,?,?,?,?,?,?,?,?,?)");
    }
    MySQL analisis = new MySQL();
    // Variables declaration - do not modify
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel12;
    private javax.swing.JLabel jLabel13;
    private javax.swing.JLabel jLabel15;
    private javax.swing.JLabel jLabel16;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel31;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
```

```
private javax.swing.JLabel jLabel9;  
public static javax.swing.JPanel jPanel1;  
private javax.swing.JPanel jPanel2;  
private javax.swing.JScrollPane jScrollPane1;  
public static javax.swing.JTabbedPane jTabbedPane1;  
private javax.swing.JButton jbBuscar1;  
private javax.swing.JButton jbBuscar2;  
private javax.swing.JButton jbRegistrar;  
private javax.swing.JLabel jlAnálisisAnteriores;  
public static javax.swing.JLabel jlCodAnimal;  
public static javax.swing.JLabel jlDniDoctor;  
public static javax.swing.JLabel jlDniDueño;  
private javax.swing.JLabel jlFecha;  
public static javax.swing.JLabel jlFichaClinica;  
public static javax.swing.JLabel jlIDAnimal;  
public static javax.swing.JLabel jlPropietario;  
private javax.swing.JLabel jlProtocolo;  
public static javax.swing.JLabel jlResponsable;  
private javax.swing.JLabel jlResponsable1;  
private javax.swing.JTextArea jtaObservaciones;  
private javax.swing.JTextField jtfAlbuminemia;  
private javax.swing.JTextField jtfCPK;  
private javax.swing.JTextField jtfCreatininemia;  
private javax.swing.JTextField jtfFosfatemia;  
private javax.swing.JTextField jtfGOT;  
private javax.swing.JTextField jtfGPT;  
private javax.swing.JTextField jtfGlucemia;  
private javax.swing.JTextField jtfLDH;  
private javax.swing.JTextField jtfUremia;  
// End of variables declaration  
}
```

Urianalis.Java

```
package Examenes;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import java.util.Calendar;
public class Urianalis extends javax.swing.JInternalFrame {
    public Urianalis() {
        initComponents();
        Calendar Cal= Calendar.getInstance();
        String fec= Cal.get(Calendar.DATE)+"/"+(Cal.get(Calendar.MONTH)+1)+"/"+Cal.get(Calendar.YEAR);
        jlFecha.setText(fec);
        jlCodAnimal.setVisible(false);
        jlDniDueño.setVisible(false);
        jlDniDoctor.setVisible(false);
    }
    /*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")
    private void jbBuscar1ActionPerformed(java.awt.event.ActionEvent evt) {
        BuscarCasoClinico buscar = new BuscarCasoClinico();
        PantallaPrincipal.jdpFondo.add(buscar);
        BuscarCasoClinico.jlBuscarCasoClinico.setText("3");
        buscarToFront();
        buscar.setVisible(true);
    }
    private void jbBuscar2ActionPerformed(java.awt.event.ActionEvent evt) {
        BuscarDocente buscarDocente = new BuscarDocente();
        PantallaPrincipal.jdpFondo.add(buscarDocente);
        BuscarDocente.jlBuscarDocente.setText("3");
        buscarDocenteToFront();
        buscarDocente.setVisible(true);
    }
    private void jbRegistrarActionPerformed(java.awt.event.ActionEvent evt) {
        String protocolo_lab = jlProtocolo.getText();
        String fecha = jlFecha.getText();
        String caso_clinico = jlFichaClinica.getText();
        String cod_animal = jlCodAnimal.getText();
        String dni_doctor = jlDniDoctor.getText();
        String color = jtfColor.getText();
        String aspecto = jtfAspecto.getText();
        String densidad = jtfDensidad.getText();
        String ph = jtfPH.getText();
        String proteina = jtfProteina.getText();
        String urobilinogeno = jtfUrobilinogeno.getText();
        String glucosa = jtfGlucosa.getText();
        String c_cetonicos = jtfCCetonicos.getText();
        String leucocitos = jtfLeucocitos.getText();
        String nitritos = jtfNitritos.getText();
        String sangre_oculta = jtfSangreOculta.getText();
        String pig_biliares = jtfPigBiliares.getText();
        String observaciones= jtaObservaciones1.getText();
        String celulas_sanguineas = (String)jcbCelulasSanguineas.getSelectedItem();
        String celulas_sanguineas2 = jtfCelulasSanguineas.getText();
        String celulas_epiteliales = (String)jcbCelulasEpiteliales.getSelectedItem();
        String celulas_epiteliales2 = jtfCelulasEpiteliales.getText();
        String cilindros = (String)jcbCilindros.getSelectedItem();
        String cilindros2 = jtfCilindros.getText();
        String cristales = (String)jcbCristales.getSelectedItem();
        String cristales2 = jtfCristales.getText();
        String otra = (String)jcbOtra.getSelectedItem();
        String otra2 = jtfOtra.getText();
        String observaciones2= jtaObservaciones2.getText();
        String datos[] =
        {protocolo_lab,fecha,caso_clinico,cod_animal,dni_doctor,color,aspecto,densidad,ph,proteina,urobilinogeno,glucosa,c_cetonicos,leucocitos,nitritos,s
        angre_oculta,pig_biliares,observaciones,celulas_sanguineas,celulas_sanguineas2,celulas_epiteliales,celulas_epiteliales2,cilindros,cilindros2,cristales,c
        ristales2,otra,otra2,observaciones2};
    }
}
```

```

    analisis.InsertarRegistro(datos, "insert into urianalisis values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");
}
MySQL analisis = new MySQL();
// Variables declaration - do not modify
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel17;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel21;
private javax.swing.JLabel jLabel22;
private javax.swing.JLabel jLabel23;
private javax.swing.JLabel jLabel24;
private javax.swing.JLabel jLabel25;
private javax.swing.JLabel jLabel26;
private javax.swing.JLabel jLabel27;
private javax.swing.JLabel jLabel28;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel31;
private javax.swing.JLabel jLabel32;
private javax.swing.JLabel jLabel33;
private javax.swing.JLabel jLabel34;
private javax.swing.JLabel jLabel35;
private javax.swing.JLabel jLabel36;
private javax.swing.JLabel jLabel37;
private javax.swing.JLabel jLabel38;
private javax.swing.JLabel jLabel39;
private javax.swing.JLabel jLabel40;
private javax.swing.JLabel jLabel41;
private javax.swing.JLabel jLabel42;
private javax.swing.JLabel jLabel43;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel9;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JTabbedPane jTabbedPane1;
private javax.swing.JTabbedPane jTabbedPane2;
private javax.swing.JButton jbBuscar1;
private javax.swing.JButton jbBuscar2;
private javax.swing.JButton jbRegistrar;
private javax.swing.JComboBox jcbCelulasEpiteliales;
private javax.swing.JComboBox jcbCelulasSanguineas;
private javax.swing.JComboBox jcbCilindros;
private javax.swing.JComboBox jcbCristales;
private javax.swing.JComboBox jcbOtra;
public static javax.swing.JLabel jlCodAnimal;
public static javax.swing.JLabel jlDniDoctor;
public static javax.swing.JLabel jlDniDueño;
private javax.swing.JLabel jlFecha;
public static javax.swing.JLabel jlFichaClinica;
public static javax.swing.JLabel jlIDAnimal;
public static javax.swing.JLabel jlPropietario;
private javax.swing.JLabel jlProtocolo;
public static javax.swing.JLabel jlResponsable;
private javax.swing.JPanel jpDatosPaciente;
private javax.swing.JPanel jpExamenDelSedimento;
private javax.swing.JPanel jpExamenFisico;
private javax.swing.JPanel jpExamenQuimico;
private javax.swing.JTextArea jtaObservaciones1;
private javax.swing.JTextArea jtaObservaciones2;
private javax.swing.JTextField jtfAspecto;
private javax.swing.JTextField jtfCCetonicos;
private javax.swing.JTextField jtfCelulasEpiteliales;
private javax.swing.JTextField jtfCelulasSanguineas;
private javax.swing.JTextField jtfCilindros;
private javax.swing.JTextField jtfColor;
private javax.swing.JTextField jtfCristales;
private javax.swing.JTextField jtfDensidad;

```



```
private javax.swing.JTextField jtfGlucosa;  
private javax.swing.JTextField jtfLeucocitos;  
private javax.swing.JTextField jtfNitritos;  
private javax.swing.JTextField jtfOtra;  
private javax.swing.JTextField jtfOtro;  
private javax.swing.JTextField jtfPH;  
private javax.swing.JTextField jtfPigBiliares;  
private javax.swing.JTextField jtfProteina;  
private javax.swing.JTextField jtfSangreOculta;  
private javax.swing.JTextField jtfUrobilinogeno;  
// End of variables declaration  
}
```

Paquete ModuloAnimal

BuscarAnimalP.Java

```
package ModuloAnimal;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import javax.swing.table.DefaultTableModel;
public class BuscarAnimalP extends javax.swing.JFrame {
    public BuscarAnimalP() {
        initComponents();
        ImgAnimalp img = new ImgAnimalp();
        jplImagen.add(img);
        jplImagen.repaint();
        cargarTabla();
    }
    private void cargarTabla(){
        String[] columnNames = {"Apellido", "Nombre", "DNI", "Nombre", "Especie", "Raza", "Peso", "Fecha", "Sexo", "Color", "Vivo", "Codigo Animal"};
        String colName[] =
{"apellido", "nombre", "dni_duenio", "animal_pequenos.nombre", "especie", "raza", "peso", "nacimiento", "sexo", "color", "vivo", "cod_animal"};
        //nombre de columnas, tabla, instruccion sql
        dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`. * FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
(`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio`) AND (`animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie`) AND
(`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza`)");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtAnimal, columnNames);
        jtAnimal.setModel(datos); // se colocan los datos en la tabla
        //depurador de jtDuenio de item null
        int a = jtAnimal.getRowCount();
        int i=0;
        a--;
        while(a>=i){
            if(jtAnimal.getValueAt(i,1)==null){
                ((DefaultTableModel)jtAnimal.getModel()).removeRow(i);
                a--;
                i--;
            }
            i++;
        }
    }
    /*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")
    private void jtAnimalMouseClicked(java.awt.event.MouseEvent evt) {
        fila=jtAnimal.rowAtPoint(evt.getPoint());
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String texto= jtfBuscar.getText();
        String[] columnNames = {"Apellido", "Nombre", "DNI", "Nombre", "Especie", "Raza", "Peso", "Fecha", "Sexo", "Color", "Vivo", "Codigo Animal"};
        String colName[] =
{"apellido", "nombre", "dni_duenio", "animal_pequenos.nombre", "especie", "raza", "peso", "nacimiento", "sexo", "color", "vivo", "cod_animal"};
        //nombre de columnas, tabla, instruccion sql
        //nombre de columnas, tabla, instruccion sql
        if(jrbDni.isSelected()==true){
            dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`. * FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio` AND `animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie` AND
`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza` AND `duenio`.`dni_duenio`="+texto+" ");
        }
        if(jrbNombre.isSelected()==true){
```



```

        dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, animal_pequenos`. *FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
(`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio`) AND (`animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie`) AND
(`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza`) AND (`animal_pequenos`.`nombre` = \" + texto + "\" )");
    }
    DefaultTableModel datos = new DefaultTableModel(dtAnimal, columnNames);
    jtAnimal.setModel(datos); // se colocan los datos en la tabla
//depurador de jtDuenio de item null
    int a = jtAnimal.getRowCount();
    int i=0;
    a--;
    while(a>=i){
        if(jtAnimal.getValueAt(i,1)==null){
            ((DefaultTableModel)jtAnimal.getModel()).removeRow(i);
            a--;
            i--;
        }
        i++;
    }
}

private void jbDetalleActionPerformed(java.awt.event.ActionEvent evt) {
    DetalleAnimal det=new DetalleAnimal();
    PantallaPrincipal.jdpFondo.add(det);
    det.show();
}

Object[][] dtAnimal;
Object[] registro;
MySQL animal = new MySQL();
int fila = -1;
// Variables declaration - do not modify
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JButton jbBuscar;
private javax.swing.JButton jbCancelar;
private javax.swing.JButton jbDetalle;
private javax.swing.JPanel jPanel3;
private javax.swing.JRadioButton jrbDni;
private javax.swing.JRadioButton jrbNombre;
public static javax.swing.JTable jtAnimal;
private javax.swing.JTextField jtfBuscar;
// End of variables declaration
}

```

DetalleAnimal.Java

```
package ModuloAnimal;
public class DetalleAnimal extends javax.swing.JInternalFrame {
    public DetalleAnimal() {
        initComponents();
        int fila = BuscarAnimalP.jtAnimal.getSelectedRow();
        jlApellido.setText(BuscarAnimalP.jtAnimal.getValueAt(fila, 0).toString());
        jlNombre.setText(BuscarAnimalP.jtAnimal.getValueAt(fila, 1).toString());
        jlDni.setText(BuscarAnimalP.jtAnimal.getValueAt(fila, 2).toString());
        jlNombreAnimal.setText(BuscarAnimalP.jtAnimal.getValueAt(fila, 3).toString());
        jlEspecie.setText(BuscarAnimalP.jtAnimal.getValueAt(fila, 4).toString());
        jlRaza.setText(BuscarAnimalP.jtAnimal.getValueAt(fila, 5).toString());
        jlPeso.setText(BuscarAnimalP.jtAnimal.getValueAt(fila, 6).toString());
        jlFecha.setText(BuscarAnimalP.jtAnimal.getValueAt(fila, 7).toString());
        jlSexo.setText(BuscarAnimalP.jtAnimal.getValueAt(fila, 8).toString());
        jlColor.setText(BuscarAnimalP.jtAnimal.getValueAt(fila, 9).toString());
        jlVivo.setText(BuscarAnimalP.jtAnimal.getValueAt(fila, 10).toString());
    }
    /*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")
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    // Variables declaration - do not modify
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JButton jbCancelar;
    private javax.swing.JLabel jlApellido;
    private javax.swing.JLabel jlColor;
    private javax.swing.JLabel jlDni;
    private javax.swing.JLabel jlEspecie;
    private javax.swing.JLabel jlFecha;
    private javax.swing.JLabel jlNombre;
    private javax.swing.JLabel jlNombreAnimal;
    private javax.swing.JLabel jlPeso;
    private javax.swing.JLabel jlRaza;
    private javax.swing.JLabel jlSexo;
    private javax.swing.JLabel jlVivo;
    // End of variables declaration
}
```

EspecieRazaAnimalG.Java

```
package ModuloAnimal;
import Clases.MySQL;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
public class EspecieRazaAnimalG extends javax.swing.JInternalFrame {
    public EspecieRazaAnimalG() {
        initComponents();
        ImgAnimalp img = new ImgAnimalp();
        jpimagen.add(img);
        jpimagen.repaint();
        cargarTabla();
        cargarEspecies();
        cargarEspecies2();
        cargarRazas();
        cargarRazas2();
    }
    private void cargarTabla(){
        if (jrbEspecie.isSelected() == true) {
            String[] columnNames = {"Codigo de Especie", "Nombre de la especie"};
            String colName[] = {"cod_especie", "especie"};
            //nombre de columnas, tabla, instruccion sql
            dtAnimal = animal.GetTabla(colName, "lista_especieg", "SELECT *FROM `lista_especieg`");
            // se colocan los datos en la tabla
            DefaultTableModel datos = new DefaultTableModel(dtAnimal, columnNames);
            jtEspecie.setModel(datos);
        }
        if (jrbEspecieRaza.isSelected() == true) {
            String[] columnNames = {"Codigo de Especie", "Nombre de la especie", "Codigo de Raza", "Nombre de la raza"};
            String colName[] = {"cod_especie", "especie", "cod_raza", "raza"};
            //nombre de columnas, tabla, instruccion sql
            dtAnimal = animal.GetTabla(colName, "lista_especieg, lista_razag", "SELECT *FROM `lista_especieg`,`lista_razag`WHERE `lista_especieg`.`cod_especie`=`lista_razag`.`cod_especie`");
            // se colocan los datos en la tabla
            DefaultTableModel datos = new DefaultTableModel(dtAnimal, columnNames);
            jtEspecie.setModel(datos);
            // se colocan los datos en la tabla
            int a = jtEspecie.getRowCount();
            //depurador de jtDuenio de item null
            int i=0;
            a--;
            while(a>=i){
                if(jtEspecie.getValueAt(i,1)==null){
                    ((DefaultTableModel)jtEspecie.getModel()).removeRow(i);
                    a--;
                    i--;
                }
                i++;
            }
        }
    }
    private void desbloquearAgregar(){
        jcbEspecie.setEnabled(true);
        jcbRaza.setEnabled(true);
        jbActualizarRaza.setEnabled(false);
        jbActualizarEspecie.setEnabled(false);
        jbCrearRaza.setEnabled(false);
        jbAgregarRaza.setEnabled(true);
        jbAgregarEspecie.setEnabled(true);
        jbAgregarEspecie2.setEnabled(false);
        jbAgregarRaza2.setEnabled(false);
        jtfEspecie.setEnabled(false);
        jtfRaza.setEnabled(false);
    }
    private void desbloquearModificar(){
        jcbEspecie.setEnabled(false);
    }
}
```

```

jcbRaza.setEnabled(false);
jbAgregaRaza.setEnabled(false);
jbCreaRaza.setEnabled(true);
jbAgregaEspecie2.setEnabled(true);
jbAgregaEspecie.setEnabled(false);
jtfEspecie.setEnabled(true);
jbActualizarEspecie.setEnabled(true);
if (jrbEspecieRaza.isSelected() == true) {
    jtfRaza.setEnabled(true);
    jbAgregaRaza2.setEnabled(true);
    jbActualizarRaza.setEnabled(true);
}else{
    jtfRaza.setEnabled(false);
    jbAgregaRaza2.setEnabled(false);
    jbActualizarRaza.setEnabled(false);
}
}
private void cargarEspecies(){
//nombre de columnas, tabla, instruccion sql
dcEspecie = animal.GetColumna("lista_especieg", "especie", "SELECT * FROM `lista_especieg`");
// se colocan los datos en la tabla
DefaultComboBoxModel datos = new DefaultComboBoxModel(dcEspecie);
jcbEspecie.setModel(datos);
}
private void cargarEspecies2(){
//nombre de columnas, tabla, instruccion sql
dcEspecie = animal.GetColumna("lista_especieg", "especie", "SELECT * FROM `lista_especieg`");
// se colocan los datos en la tabla
DefaultComboBoxModel datos = new DefaultComboBoxModel(dcEspecie);
jcbEspecie2.setModel(datos);
}
private void cargarRazas(){
jcbRaza.removeAllItems();
String especie = jcbEspecie.getSelectedItem().toString();
String cod_especie = animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especieg WHERE especie = \""+especie+"\"");
//nombre de columnas, tabla, instruccion sql
dcRaza = animal.GetColumna("lista_razag", "raza", "SELECT * FROM lista_razag WHERE cod_especie = \""+cod_especie+"\"");
// se colocan los datos en la tabla
DefaultComboBoxModel datos = new DefaultComboBoxModel(dcRaza);
jcbRaza.setModel(datos);
int a = jcbRaza.getItemCount();
//depurador de jcbRaza de item null
int i=0;
a--;
while(a>=i){
    if(jcbRaza.getItemAt(i)==null){
        jcbRaza.removeItemAt(i);
        a--;
        i--;
    }
    i++;
}
}
private void cargarRazas2(){
jcbRaza2.removeAllItems();
String especie = jcbEspecie2.getSelectedItem().toString();
String cod_especie = animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especieg WHERE especie = \""+especie+"\"");
//nombre de columnas, tabla, instruccion sql
dcRaza = animal.GetColumna("lista_razag", "raza", "SELECT * FROM lista_razag WHERE cod_especie = \""+cod_especie+"\"");
// se colocan los datos en la tabla
DefaultComboBoxModel datos = new DefaultComboBoxModel(dcRaza);
jcbRaza2.setModel(datos);
int a = jcbRaza2.getItemCount();
//depurador de jcbRaza de item null
int i=0;
a--;
while(a>=i){
    if(jcbRaza2.getItemAt(i)==null){
        jcbRaza2.removeItemAt(i);
        a--;
        i--;
    }
    i++;
}
}

```

```

    }
    i++;
}
}
/*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")
private void jmiModificarEspecieActionPerformed(java.awt.event.ActionEvent evt) {
    int filamodificar = jtEspecie.getSelectedRow();
    if (jrbEspecie.isSelected() == true) {
        desbloquearModificar();
        if (filamodificar >= 0) {
            jlCodigoEspecie.setText(jtEspecie.getValueAt(filamodificar, 0).toString());
            jcbEspecie2.setSelectedItem(jtEspecie.getValueAt(filamodificar, 1).toString());
            jtfEspecie.setText(jtEspecie.getValueAt(filamodificar, 1).toString());
        } else {
            JOptionPane.showMessageDialog(this, "No ha seleccionado ");
        }
    }
    if (jrbEspecieRaza.isSelected() == true) {
        if (filamodificar >= 0) {
            desbloquearModificar();
            jlCodigoEspecie.setText(jtEspecie.getValueAt(filamodificar, 0).toString());
            jcbEspecie2.setSelectedItem(jtEspecie.getValueAt(filamodificar, 1).toString());
            jtfEspecie.setText(jtEspecie.getValueAt(filamodificar, 1).toString());
            jlCodigoRaza.setText(jtEspecie.getValueAt(filamodificar, 2).toString());
            jcbRaza2.setSelectedItem(jtEspecie.getValueAt(filamodificar, 3).toString());
            jtfRaza.setText(jtEspecie.getValueAt(filamodificar, 3).toString());
        } else {
            JOptionPane.showMessageDialog(this, "No ha seleccionado ");
        }
    }
}
private void jbCrearRazaActionPerformed(java.awt.event.ActionEvent evt) {
    desbloquearAgregar();
}
private void jbAgregarRazaActionPerformed(java.awt.event.ActionEvent evt) {
    String especie= (String)jcbEspecie.getSelectedItem();
    String cod_especie = animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especieg WHERE especie = '\"'+especie+'\"'");
    String nuevараза = JOptionPane.showInputDialog(rootPane, "Ingrese una raza para "+especie+"", "Nueva Raza", WIDTH);
    boolean aux = animal.existe("lista_razag", "raza", nuevараза);
    if (aux==true){
        JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
    }else{
        String dato[] = {cod_especie, nuevараза};
        animal.InsertarRegistro(dato, "insert into lista_razag (`cod_especie`, `raza`) values(?,?)");
        cargarRazas();
        cargarRazas2();
    }
}
private void jbActualizarRazaActionPerformed(java.awt.event.ActionEvent evt) {
    String raza = jtfRaza.getText();
    String cod_raza = jlCodigoRaza.getText();
    String datos[] = {raza, cod_raza};
    animal.UpdateRegistro(datos, "UPDATE `lista_razag` SET `raza`=? WHERE `cod_raza`=?");
    cargarTabla();
    cargarRazas();
    cargarRazas2();
}
private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}
private void jbAgregarEspecieActionPerformed(java.awt.event.ActionEvent evt) {
    String nuevaesp = JOptionPane.showInputDialog(rootPane, "Ingrese la especie que desea agregar:", "Nueva Especie", WIDTH);
    boolean aux = animal.existe("lista_especieg", "especie", nuevaesp);
    if (aux==true){
        JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
    }else{
        String datos[] = {nuevaesp};
        //array de datos , instruccion sql

```

```

        animal.InsertarRegistro(datos, "insert into lista_especieg (`especie`)values(?)");
        cargarEspecies();
        cargarEspecies2();
    }
}
private void jcbEspecieItemStateChanged(java.awt.event.ItemEvent evt) {
    cargarRazas();
}
private void jcbEspecie2ItemStateChanged(java.awt.event.ItemEvent evt) {
    cargarRazas2();
}
private void jbAgregarEspecie2ActionPerformed(java.awt.event.ActionEvent evt) {
    String nuevaesp = JOptionPane.showInputDialog(rootPane,"Ingrese la especie que desea agregar:", "Nueva Especie", WIDTH);
    boolean aux = animal.existe("lista_especieg", "especie", nuevaesp);
    if (aux==true){
        JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
    }else{
        String datos[] = {nuevaesp};
        //array de datos , instruccion sql
        animal.InsertarRegistro(datos, "insert into lista_especieg (`especie`)values(?)");
        cargarEspecies();
        cargarEspecies2();
    }
}
private void jbAgregarRaza2ActionPerformed(java.awt.event.ActionEvent evt) {
    String especie= (String)jcbEspecie2.getSelectedItem();
    String cod_especie = animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especieg WHERE especie = \"\""+especie+"\"");
    String nuevараза = JOptionPane.showInputDialog(rootPane,"Ingrese una raza para "+especie+"", "Nueva Raza", WIDTH);
    boolean aux = animal.existe("lista_razag", "raza", nuevараза);
    if (aux==true){
        JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
    }else{
        String dato[] = {cod_especie,nuevараза};
        animal.InsertarRegistro(dato, "insert into lista_razag (`cod_especie`,`raza`)values(?,?)");
        cargarRazas();
        cargarRazas2();
    }
}
private void jbActualizarEspecieActionPerformed(java.awt.event.ActionEvent evt) {
    String especie = jtfEspecie.getText();
    String cod_especie = jCodigoEspecie.getText();
    String datos[] = {especie,cod_especie};
    animal.UpdateRegistro(datos, "UPDATE `lista_especieg`SET`especie`= ? WHERE `cod_especie`= ?");
    cargarTabla();
    cargarEspecies();
    cargarEspecies2();
}
private void jrbEspecieActionPerformed(java.awt.event.ActionEvent evt) {
    cargarTabla();
}
private void jrbEspecieRazaActionPerformed(java.awt.event.ActionEvent evt) {
    cargarTabla();
}
Object[][] dtAnimal;
MySQL animal = new MySQL();
Object[] dcEspecie;
Object[] dcRaza;

// Variables declaration - do not modify
private javax.swing.ButtonGroup buttonGroup1;
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.JLabel jLabel6;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel4;
private javax.swing.JPopupMenu jPopupMenu1;

```



```
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JButton jbActualizarEspecie;  
private javax.swing.JButton jbActualizarRaza;  
private javax.swing.JButton jbAgregarEspecie;  
private javax.swing.JButton jbAgregarEspecie2;  
private javax.swing.JButton jbAgregarRaza;  
private javax.swing.JButton jbAgregarRaza2;  
private javax.swing.JButton jbCancelar;  
private javax.swing.JButton jbCrearRaza;  
private javax.swing.JComboBox jcbEspecie;  
private javax.swing.JComboBox jcbEspecie2;  
private javax.swing.JComboBox jcbRaza;  
private javax.swing.JComboBox jcbRaza2;  
private javax.swing.JLabel jlCodigoEspecie;  
private javax.swing.JLabel jlCodigoRaza;  
private javax.swing.JMenuItem jmiModificarEspecie;  
private javax.swing.JPanel jpimagen;  
private javax.swing.JRadioButton jrbEspecie;  
private javax.swing.JRadioButton jrbEspecieRaza;  
private javax.swing.JTable jtEspecie;  
private javax.swing.JTextField jtfEspecie;  
private javax.swing.JTextField jtfRaza;  
// End of variables declaration  
}
```


EspecieRazaAnimalP.Java

```
package ModuloAnimal;
import Clases.MySQL;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
public class EspecieRazaAnimalP extends javax.swing.JInternalFrame {
    public EspecieRazaAnimalP() {
        initComponents();
        ImgAnimalp img = new ImgAnimalp();
        jpimagen.add(img);
        jpimagen.repaint();
        cargarTabla();
        cargarEspecies();
        cargarEspecies2();
        cargarRazas();
        cargarRazas2();
    }
    private void cargarTabla(){
        if (jrbEspecie.isSelected() == true) {
            String[] columnNames = {"Codigo de Especie", "Nombre de la especie"};
            String colName[] = {"cod_especie", "especie"};
            //nombre de columnas, tabla, instruccion sql
            dtAnimal = animal.GetTabla(colName, "lista_especie", "SELECT *FROM `lista_especie`");
            // se colocan los datos en la tabla
            DefaultTableModel datos = new DefaultTableModel(dtAnimal, columnNames);
            jtEspecie.setModel(datos);
        }
        if (jrbEspecieRaza.isSelected() == true) {
            String[] columnNames = {"Codigo de Especie", "Nombre de la especie", "Codigo de Raza", "Nombre de la raza"};
            String colName[] = {"cod_especie", "especie", "cod_raza", "raza"};
            //nombre de columnas, tabla, instruccion sql
            dtAnimal = animal.GetTabla(colName, "lista_especie, lista_raza", "SELECT *FROM `lista_especie`, `lista_raza` WHERE `lista_especie`.`cod_especie` = `lista_raza`.`cod_especie`");
            // se colocan los datos en la tabla
            DefaultTableModel datos = new DefaultTableModel(dtAnimal, columnNames);
            jtEspecie.setModel(datos);
            // se colocan los datos en la tabla
            int a = jtEspecie.getRowCount();
            //depurador de jtDuenio de item null
            int i=0;
            a--;
            while(a>=i){
                if(jtEspecie.getValueAt(i,1)==null){
                    ((DefaultTableModel)jtEspecie.getModel()).removeRow(i);
                    a--;
                    i--;
                }
                i++;
            }
        }
    }
    private void desbloquearAgregar(){
        jcbEspecie.setEnabled(true);
        jcbRaza.setEnabled(true);
        jbActualizarRaza.setEnabled(false);
        jbActualizarEspecie.setEnabled(false);
        jbCrearRaza.setEnabled(false);
        jbAgregarRaza.setEnabled(true);
        jbAgregarEspecie.setEnabled(true);
        jbAgregarEspecie2.setEnabled(false);
        jbAgregarRaza2.setEnabled(false);
        jtfEspecie.setEnabled(false);
        jtfRaza.setEnabled(false);
    }
    private void desbloquearModificar(){
        jcbEspecie.setEnabled(false);
    }
}
```

```

jcbRaza.setEnabled(false);
jbAgregaRaza.setEnabled(false);
jbCrearRaza.setEnabled(true);
jbAgregaEspecie2.setEnabled(true);
jbAgregaEspecie.setEnabled(false);
jtfEspecie.setEnabled(true);
jbActualizarEspecie.setEnabled(true);
if (jrbEspecieRaza.isSelected() == true) {
    jtfRaza.setEnabled(true);
    jbAgregaRaza2.setEnabled(true);
    jbActualizarRaza.setEnabled(true);
}else{
    jtfRaza.setEnabled(false);
    jbAgregaRaza2.setEnabled(false);
    jbActualizarRaza.setEnabled(false);
}
}
private void cargarEspecies(){
//nombre de columnas, tabla, instruccion sql
dcEspecie = animal.GetColumna("lista_especie", "especie", "SELECT * FROM `lista_especie`");
// se colocan los datos en la tabla
DefaultComboBoxModel datos = new DefaultComboBoxModel(dcEspecie);
jcbEspecie.setModel(datos);
}
private void cargarEspecies2(){
//nombre de columnas, tabla, instruccion sql
dcEspecie = animal.GetColumna("lista_especie", "especie", "SELECT * FROM `lista_especie`");
// se colocan los datos en la tabla
DefaultComboBoxModel datos = new DefaultComboBoxModel(dcEspecie);
jcbEspecie2.setModel(datos);
}
private void cargarRazas(){
jcbRaza.removeAllItems();
String especie = jcbEspecie.getSelectedItem().toString();
String cod_especie = animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especie WHERE especie = \""+especie+"\"");
//nombre de columnas, tabla, instruccion sql
dcRaza = animal.GetColumna("lista_raza", "raza", "SELECT * FROM lista_raza WHERE cod_especie = \""+cod_especie+"\"");
// se colocan los datos en la tabla
DefaultComboBoxModel datos = new DefaultComboBoxModel(dcRaza);
jcbRaza.setModel(datos);
int a = jcbRaza.getItemCount();
//depurador de jcbRaza de item null
int i=0;
a--;
while(a>=i){
    if(jcbRaza.getItemAt(i)==null){
        jcbRaza.removeItemAt(i);
        a--;
        i--;
    }
    i++;
}
}
private void cargarRazas2(){
jcbRaza2.removeAllItems();
String especie = jcbEspecie2.getSelectedItem().toString();
String cod_especie = animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especie WHERE especie = \""+especie+"\"");
//nombre de columnas, tabla, instruccion sql
dcRaza = animal.GetColumna("lista_raza", "raza", "SELECT * FROM lista_raza WHERE cod_especie = \""+cod_especie+"\"");
// se colocan los datos en la tabla
DefaultComboBoxModel datos = new DefaultComboBoxModel(dcRaza);
jcbRaza2.setModel(datos);
int a = jcbRaza2.getItemCount();
//depurador de jcbRaza de item null
int i=0;
a--;
while(a>=i){
    if(jcbRaza2.getItemAt(i)==null){
        jcbRaza2.removeItemAt(i);
        a--;
        i--;
    }
    i++;
}
}

```

```

    }
    i++;
}
}
/*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")
private void jmiModificarEspecieActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int filamodificar = jtEspecie.getSelectedRow();
    if (jrbEspecie.isSelected() == true) {
        desbloquearModificar();
        if (filamodificar >= 0) {
            jlCodigoEspecie.setText(jtEspecie.getValueAt(filamodificar, 0).toString());
            jcbEspecie2.setSelectedItem(jtEspecie.getValueAt(filamodificar, 1).toString());
            jtfEspecie.setText(jtEspecie.getValueAt(filamodificar, 1).toString());
        } else {
            JOptionPane.showMessageDialog(this, "No ha seleccionado ");
        }
    }
    if (jrbEspecieRaza.isSelected() == true) {
        if (filamodificar >= 0) {
            desbloquearModificar();
            jlCodigoEspecie.setText(jtEspecie.getValueAt(filamodificar, 0).toString());
            jcbEspecie2.setSelectedItem(jtEspecie.getValueAt(filamodificar, 1).toString());
            jtfEspecie.setText(jtEspecie.getValueAt(filamodificar, 1).toString());
            jlCodigoRaza.setText(jtEspecie.getValueAt(filamodificar, 2).toString());
            jcbRaza2.setSelectedItem(jtEspecie.getValueAt(filamodificar, 3).toString());
            jtfRaza.setText(jtEspecie.getValueAt(filamodificar, 3).toString());
        } else {
            JOptionPane.showMessageDialog(this, "No ha seleccionado ");
        }
    }
}
private void jbCrearRazaActionPerformed(java.awt.event.ActionEvent evt) {
    desbloquearAgregar();
}
private void jbAgregarRazaActionPerformed(java.awt.event.ActionEvent evt) {
    String especie= (String)jcbEspecie.getSelectedItem();
    String cod_especie = animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especie WHERE especie = '"+especie+"'");
    String nuevараза = JOptionPane.showInputDialog(rootPane,"Ingrese una raza para "+especie+"","Nueva Raza",WIDTH);
    boolean aux = animal.existe("lista_raza", "raza", nuevараза);
    if (aux==true){
        JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
    }else{
        String dato[] = {cod_especie,nuevараза};
        animal.InsertarRegistro(dato, "insert into lista_raza (`cod_especie`,`raza`)values(?,?)");
        cargarRazas();
        cargarRazas2();
    }
}
private void jbActualizarRazaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String raza = jtfRaza.getText();
    String cod_raza = jlCodigoRaza.getText();
    String datos[] = {raza,cod_raza};
    animal.UpdateRegistro(datos, "UPDATE `lista_raza`SET`raza`= ? WHERE `cod_raza`= ?");
    cargarTabla();
    cargarRazas();
    cargarRazas2();
}
private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}
private void jbAgregarEspecieActionPerformed(java.awt.event.ActionEvent evt) {
    String nuevaesp = JOptionPane.showInputDialog(rootPane,"Ingrese la especie que desea agregar:","Nueva Especie", WIDTH);
    boolean aux = animal.existe("lista_especie", "especie", nuevaesp);
    if (aux==true){
        JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
    }else{

```

```

        String datos[] = {nuevaesp};
        //array de datos , instruccion sql
        animal.InsertarRegistro(datos, "insert into lista_especie (`especie`)values(?)");
        cargarEspecies();
        cargarEspecies2();
    }
}

private void jcbEspecieItemStateChanged(java.awt.event.ItemEvent evt) {
    cargarRazas();
}

private void jcbEspecie2ItemStateChanged(java.awt.event.ItemEvent evt) {
    cargarRazas2();
}

private void jbAgregarEspecie2ActionPerformed(java.awt.event.ActionEvent evt) {
    String nuevaesp = JOptionPane.showInputDialog(rootPane,"Ingrese la especie que desea agregar:", "Nueva Especie", WIDTH);
    boolean aux = animal.existe("lista_especie", "especie", nuevaesp);
    if (aux==true){
        JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
    }else{
        String datos[] = {nuevaesp};
        //array de datos , instruccion sql
        animal.InsertarRegistro(datos, "insert into lista_especie (`especie`)values(?)");
        cargarEspecies();
        cargarEspecies2();
    }
}

private void jbAgregarRaza2ActionPerformed(java.awt.event.ActionEvent evt) {
    String especie= (String)jcbEspecie2.getSelectedItem();
    String cod_especie = animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especie WHERE especie = \"\""+especie+"\"");
    String nuevaraza = JOptionPane.showInputDialog(rootPane,"Ingrese una raza para "+especie+"", "Nueva Raza", WIDTH);
    boolean aux = animal.existe("lista_raza", "raza", nuevaraza);
    if (aux==true){
        JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
    }else{
        String dato[] = {cod_especie,nuevaraza};
        animal.InsertarRegistro(dato, "insert into lista_raza (`cod_especie`,`raza`)values(?,?)");
        cargarRazas();
        cargarRazas2();
    }
}

private void jbActualizarEspecieActionPerformed(java.awt.event.ActionEvent evt) {
    String especie = jtfEspecie.getText();
    String cod_especie = jCodigoEspecie.getText();
    String datos[] = {especie,cod_especie};
    animal.UpdateRegistro(datos, "UPDATE `lista_especie` SET `especie`=? WHERE `cod_especie`=?");
    cargarTabla();
    cargarEspecies();
    cargarEspecies2();
}

private void jrbEspecieActionPerformed(java.awt.event.ActionEvent evt) {
    cargarTabla();
}

private void jrbEspecieRazaActionPerformed(java.awt.event.ActionEvent evt) {
    cargarTabla();
}

Object[][] dtAnimal;
MySQL animal = new MySQL();
Object[] dcEspecie;
Object[] dcRaza;
// Variables declaration - do not modify
private javax.swing.ButtonGroup buttonGroup1;
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.JLabel jLabel6;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel4;

```

```
private javax.swing.JPopupMenu jPopupMenu1;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JButton jbActualizarEspecie;  
private javax.swing.JButton jbActualizarRaza;  
private javax.swing.JButton jbAgregarEspecie;  
private javax.swing.JButton jbAgregarEspecie2;  
private javax.swing.JButton jbAgregarRaza;  
private javax.swing.JButton jbAgregarRaza2;  
private javax.swing.JButton jbCancelar;  
private javax.swing.JButton jbCrearRaza;  
private javax.swing.JComboBox jcbEspecie;  
private javax.swing.JComboBox jcbEspecie2;  
private javax.swing.JComboBox jcbRaza;  
private javax.swing.JComboBox jcbRaza2;  
private javax.swing.JLabel jlCodigoEspecie;  
private javax.swing.JLabel jlCodigoRaza;  
private javax.swing.JMenuItem jmiModificarEspecie;  
private javax.swing.JPanel jpimagen;  
private javax.swing.JRadioButton jrbEspecie;  
private javax.swing.JRadioButton jrbEspecieRaza;  
private javax.swing.JTable jtEspecie;  
private javax.swing.JTextField jtfEspecie;  
private javax.swing.JTextField jtfRaza;  
// End of variables declaration  
}
```

ImgAnimalp.Java

```
package ModuloAnimal;
import ModuloDueño.*;
import java.awt.Graphics;
import javax.swing.ImageIcon;
public class ImgAnimalp extends javax.swing.JPanel {
    public ImgAnimalp(){
        this.setSize(128,128);
    }
    @Override
    public void paint(Graphics g){
        ImageIcon imagenFondo = new ImageIcon(getClass().getResource("/imagenes/animalp.png"));
        g.drawImage(imagenFondo.getImage(),0,0,128,128, null);
        setOpaque(false);
        super.paintComponent(g);
    }
}
```

ListaDueño.Java

```
package ModuloAnimal;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import ModuloDueño.ImgDuenio;
import javax.swing.JFrame;
import javax.swing.JLabel;
import ModuloAnimal.NuevoAnimalP;
import ModuloDueño.NuevoDueño;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
public class ListaDueño extends javax.swing.JInternalFrame {
    public ListaDueño() {
        initComponents();
        ImgDuenio img = new ImgDuenio ();
        jpimagen.add(img);
        jpimagen.repaint();
        cargarTabla();
    }
    public void cargarTabla(){
        String[] columnNames = {"Apellido","Nombre","DNI"};
        String colName[] = {"apellido","nombre","dni_duenio"};
        //nombre de columnas, tabla, instruccion sql
        dtDuenio = duenio.GetTabla(colName, "duenio", "SELECT `dni_duenio`, `nombre`, `apellido` FROM `duenio` ORDER BY `apellido`");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtDuenio,columnNames);
        jtDueño.setModel(datos);
    }
    /*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")
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar= jtDueño.getSelectedRow();
        if(filamodificar>=0){
            String apellido= jtDueño.getValueAt(filamodificar, 0).toString();
            String nombre= jtDueño.getValueAt(filamodificar, 1).toString();
            String dni= jtDueño.getValueAt(filamodificar, 2).toString();
            if(jLabel1.getText()=="1"){
                NuevoAnimalP.jlApellido.setText(apellido);
                NuevoAnimalP.jlNombre.setText(nombre);
                NuevoAnimalP.jlDni.setText(dni);        //mensaje de que está abierto si se desea
            }
            if(jLabel1.getText()=="2"){
                ModificarAnimalP.jlApellido.setText(apellido);
                ModificarAnimalP.jlNombre.setText(nombre);
                ModificarAnimalP.jlDni.setText(dni);
            }
            ListaDueño.jLabel1.setText("0");
            this.dispose();
        }else{
            JOptionPane.showMessageDialog(this,"No ha seleccionado ");
        }
    }
    private void jtDueñoMouseClicked(java.awt.event.MouseEvent evt) {
        fila = jtDueño.rowAtPoint(evt.getPoint());
    }
    private void jbAgregarActionPerformed(java.awt.event.ActionEvent evt) {
        NuevoDueño dueño = new NuevoDueño();
        PantallaPrincipal.jdpFondo.add(dueño);
        this.dispose();
        dueño.show();
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    MySQL duenio = new MySQL();
}
```




```
Object[][] dtDuenio;  
int fila = -1;  
// Variables declaration - do not modify  
public static javax.swing.JLabel jLabel1;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JPanel jPanel2;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JButton jbAgregar;  
private javax.swing.JButton jbCancelar;  
private javax.swing.JButton jbElegir;  
private javax.swing.JPanel jpimagen;  
private javax.swing.JTable jtDueño;  
// End of variables declaration  
}
```

ListaDueñoG.Java

```

package ModuloAnimal;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import ModuloDueño.ImgDuenio;
import javax.swing.JFrame;
import javax.swing.JLabel;
import ModuloAnimal.NuevoAnimalG;
import ModuloDueño.NuevoDueño;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
public class ListaDueñoG extends javax.swing.JInternalFrame {
    public ListaDueñoG() {
        initComponents();
        ImgDuenio img = new ImgDuenio ();
        jpimagen.add(img);
        jpimagen.repaint();
        cargarTabla();
    }
    public void cargarTabla(){
        String[] columnNames = {"Apellido","Nombre","DNI"};
        String colName[] = {"apellido","nombre","dni_duenio"};
        //nombre de columnas, tabla, instruccion sql
        dtDuenio = duenio.GetTabla(colName, "duenio", "SELECT `dni_duenio`, `nombre`, `apellido` FROM `duenio` ORDER BY `apellido`");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtDuenio,columnNames);
        jtDueño.setModel(datos);
    }
    /*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")
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar= jtDueño.getSelectedRow();
        if(filamodificar>=0){
            String apellido= jtDueño.getValueAt(filamodificar, 0).toString();
            String nombre= jtDueño.getValueAt(filamodificar, 1).toString();
            String dni= jtDueño.getValueAt(filamodificar, 2).toString();
            if(jLabel1.getText()=="1"){
                NuevoAnimalG.jlApellido.setText(apellido);
                NuevoAnimalG.jlNombre.setText(nombre);
                NuevoAnimalG.jlDni.setText(dni);
            }
            if(jLabel1.getText()=="2"){
                ModificarAnimalG.jlApellido.setText(apellido);
                ModificarAnimalG.jlNombre.setText(nombre);
                ModificarAnimalG.jlDni.setText(dni);
            }
            ListaDueñoG.jLabel1.setText("0");
            this.dispose();
        }else{
            JOptionPane.showMessageDialog(this,"No ha seleccionado ");
        }
    }
    private void jtDueñoMouseClicked(java.awt.event.MouseEvent evt) {
        fila = jtDueño.rowAtPoint(evt.getPoint());
    }
    private void jbAgregarActionPerformed(java.awt.event.ActionEvent evt) {
        NuevoDueño dueño = new NuevoDueño();
        PantallaPrincipal.jdpFondo.add(dueño);
        this.dispose();
        dueño.show();
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    MySQL duenio = new MySQL();

```



```
Object[][] dtDuenio;  
int fila = -1;  
// Variables declaration - do not modify  
public static javax.swing.JLabel jLabel1;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JPanel jPanel2;  
private javax.swing.JScrollPane jScrollPane1;  
private javax.swing.JButton jbAgregar;  
private javax.swing.JButton jbCancelar;  
private javax.swing.JButton jbElegir;  
private javax.swing.JPanel jpimagen;  
private javax.swing.JTable jtDueño;  
// End of variables declaration  
}
```

ModificarAnimalG.Java

```
package ModuloAnimal;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
public class ModificarAnimalG extends javax.swing.JInternalFrame {
    public ModificarAnimalG() {
        initComponents();
        ImgAnimalp img = new ImgAnimalp();
        jplImagen.add(img);
        jplImagen.repaint();
        cargarTabla();
        cargarEspecies();
        cargarRazas();
    }
    private void cargarTabla(){
        String[] columnNames = {"Apellido", "Nombre", "DNI", "Nombre", "Especie", "Raza", "Peso", "Fecha", "Sexo", "Color", "Vivo", "Codigo Animal"};
        String colName[] =
{"apellido", "nombre", "dni_duenio", "animal_grandes.nombre", "especie", "raza", "peso", "nacimiento", "sexo", "color", "vivo", "cod_animal"};
        //nombre de columnas, tabla, instruccion sql
        dtAnimal = animal.GetTabla(colName, "animal_grandes, duenio, lista_especieg, lista_razag", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especieg`.`especie`, `lista_razag`.`raza`, `animal_grandes`. * FROM animal_grandes, duenio, lista_especieg, lista_razag WHERE
(`duenio`.`dni_duenio` = `animal_grandes`.`dni_duenio`) AND (`animal_grandes`.`cod_especie` = `lista_especieg`.`cod_especie`) AND
(`animal_grandes`.`cod_raza` = `lista_razag`.`cod_raza`)");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtAnimal, columnNames);
        jtDuenio.setModel(datos);
        // se colocan los datos en la tabla
        int a = jtDuenio.getRowCount();
        //depurador de jtDuenio de item null
        int i=0;
        a--;
        while(a>=i){
            if(jtDuenio.getValueAt(i,1)==null){
                ((DefaultTableModel)jtDuenio.getModel()).removeRow(i);
                a--;
                i--;
            }
            i++;
        }
    }
    void desbloquear(){
        jbBuscarDueño.setEnabled(true);
        jtnombre.setEnabled(true);
        jcbEspecie.setEnabled(true);
        jcbRaza.setEnabled(true);
        jbNuevaEspecie.setEnabled(true);
        jbNuevaRaza.setEnabled(true);
        jcbSexo.setEnabled(true);
        jDateChooser1.setEnabled(true);
        jtfPeso.setEnabled(true);
        jtfColor.setEnabled(true);
        jbActualizar.setEnabled(true);
    }
    private void cargarEspecies(){
        //nombre de columnas, tabla, instruccion sql
        dcEspecie = animal.GetColumna("lista_especieg", "especie", "SELECT * FROM `lista_especieg`");
        // se colocan los datos en la tabla
        DefaultComboBoxModel datos = new DefaultComboBoxModel(dcEspecie);
        jcbEspecie.setModel(datos);
    }
}
```

```

private void cargarRazas(){
    jcbRaza.removeAllItems();
    String especie = jcbEspecie.getSelectedItem().toString();
    String cod_especie = animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especieg WHERE especie = \"\""+especie+"\"");
    //nombre de columnas, tabla, instruccion sql
    dcRaza = animal.GetColumna("lista_razag", "raza", "SELECT * FROM lista_razag WHERE cod_especie = \"\""+cod_especie+"\"");
    // se colocan los datos en la tabla
    DefaultComboBoxModel datos = new DefaultComboBoxModel(dcRaza);
    jcbRaza.setModel(datos);
    int a = jcbRaza.getItemCount();
    //depurador de jcbRaza de item null
    int i=0;
    a--;
    while(a>=i){
        if(jcbRaza.getItemAt(i)==null){
            jcbRaza.removeItemAt(i);
            a--;
            i--;
        }
        i++;
    }
}

/*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")
private void jcbEspecieItemStateChanged(java.awt.event.ItemEvent evt) {
    cargarRazas();
}
private void jcbEspecieActionPerformed(java.awt.event.ActionEvent evt) {
}
private void jbNuevaEspecieActionPerformed(java.awt.event.ActionEvent evt) {
    String nuevaesp = JOptionPane.showInputDialog(rootPane,"Ingrese la especie que desea agregar:", "Nueva Especie", WIDTH);
    boolean aux = animal.existe("lista_especieg", "especie", nuevaesp);
    if (aux==true){
        JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
    }else{
        String datos[] = {nuevaesp};
        //array de datos , instruccion sql
        animal.InsertarRegistro(datos, "insert into lista_especieg (`especie`)values(?)");
        cargarEspecies();
        //buscamos el cod_especie correspondiente
        int ultimo= jcbEspecie.getItemCount()-1;
        String especie = jcbEspecie.getItemAt(ultimo).toString();
        String cod_especie = animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especieg WHERE especie = \"\""+especie+"\"");
        //agregamos la nueva raza
        String nuevараза = JOptionPane.showInputDialog(rootPane,"Ingrese una raza inicial para "+nuevaesp+"", "Nueva Raza", WIDTH);
        String dato[] = {cod_especie, nuevараза};
        animal.InsertarRegistro(dato, "insert into lista_razag (`cod_especie`,`raza`)values(?,?)");
        cargarRazas();
    }
}
private void jbNuevaRazaActionPerformed(java.awt.event.ActionEvent evt) {
    String especie= (String)jcbEspecie.getSelectedItem();
    String cod_especie = animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especieg WHERE especie = \"\""+especie+"\"");
    String nuevараза = JOptionPane.showInputDialog(rootPane,"Ingrese una raza para "+especie+"", "Nueva Raza", WIDTH);
    boolean aux = animal.existe("lista_razag", "raza", nuevараза);
    if (aux==true){
        JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
    }else{
        String dato[] = {cod_especie, nuevараза};
        animal.InsertarRegistro(dato, "insert into lista_razag (`cod_especie`,`raza`)values(?,?)");
        cargarRazas();
    }
}
private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}
private void jbActualizarActionPerformed(java.awt.event.ActionEvent evt) {
    String nombre = jtnombre.getText();
    String especie= (String)jcbEspecie.getSelectedItem();

```

```

String cod_especie = animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especieg WHERE especie = \"\"+especie+\"\"");
String raza = (String)jcbRaza.getSelectedItemAt();
String cod_raza = animal.GetData("cod_raza", "SELECT cod_raza FROM lista_razag WHERE raza = \"\"+raza+\"\"");
String peso = jtfPeso.getText();
DateFormat fecha = new SimpleDateFormat("yyyy-MM-dd");
Date nacio = jDateChooser1.getDate();
String date = fecha.format(nacio);
String sexo = (String)jcbSexo.getSelectedItemAt();
String color = jtfColor.getText();
String dni_duenio = jIDni.getText();
String life = "si";
String cod_animal = jICodAnimal.getText();
String datos[] = {nombre, cod_especie, cod_raza, peso, date, sexo, color, dni_duenio, life, cod_animal};
//array de datos, instruccion sql
animal.UpdateRegistro(datos, "UPDATE
`animal_grandes` SET `nombre`=?, `cod_especie`=?, `cod_raza`=?, `peso`=?, `nacimiento`=?, `sexo`=?, `color`=?, `dni_duenio`=?, `vivo`=? WHERE
`cod_animal`=?");
cargarTabla();
}
private void jtfbuscarMouseClicked(java.awt.event.MouseEvent evt) {
}
private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
String texto= jtfbuscar.getText();
String[] colNames = {"Apellido", "Nombre", "DNI", "Nombre", "Especie", "Raza", "Peso", "Fecha", "Sexo", "Color", "Vivo", "Codigo Animal"};
String colName[] =
{"apellido", "nombre", "dni_duenio", "animal_grandes.nombre", "especie", "raza", "peso", "nacimiento", "sexo", "color", "vivo", "cod_animal"};
//nombre de columnas, tabla, instruccion sql
//nombre de columnas, tabla, instruccion sql
if(jrbDni.isSelected()==true){
dtAnimal = animal.GetTabla(colName, "animal_grandes, duenio, lista_especieg, lista_razag", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especieg`.`especie`, `lista_razag`.`raza`, `animal_grandes`.*FROM animal_grandes, duenio, lista_especieg, lista_razag WHERE
`duenio`.`dni_duenio` = `animal_grandes`.`dni_duenio` AND `animal_grandes`.`cod_especie` = `lista_especieg`.`cod_especie` AND
`animal_grandes`.`cod_raza` = `lista_razag`.`cod_raza` AND `duenio`.`dni_duenio` = "+texto+" ");
}
if(jrbMascota.isSelected()==true){
dtAnimal = animal.GetTabla(colName, "animal_grandes, duenio, lista_especieg, lista_razag", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especieg`.`especie`, `lista_razag`.`raza`, `animal_grandes`.*FROM animal_grandes, duenio, lista_especieg, lista_razag WHERE
(`duenio`.`dni_duenio` = `animal_grandes`.`dni_duenio`) AND (`animal_grandes`.`cod_especie` = `lista_especieg`.`cod_especie`) AND
(`animal_grandes`.`cod_raza` = `lista_razag`.`cod_raza`) AND (`animal_grandes`.`nombre` = \"\" + texto + \"\" )");
}
// se colocan los datos en la tabla
DefaultTableModel datos = new DefaultTableModel(dtAnimal, colNames);
jtDuenio.setModel(datos);
int a = jtDuenio.getRowCount();
//depurador de jtDuenio de item null
int i=0;
a--;
while(a>=i){
if(jtDuenio.getValueAt(i,1)==null){
((DefaultTableModel)jtDuenio.getModel()).removeRow(i);
a--;
i--;
}
i++;
}
}
private void jmiModificarAnimalActionPerformed(java.awt.event.ActionEvent evt) {
desbloquear();
int filamodificar = jtDuenio.getSelectedRow();
if (filamodificar >= 0) {
jIApellido.setText(jtDuenio.getValueAt(filamodificar, 0).toString());
jINombre.setText(jtDuenio.getValueAt(filamodificar, 1).toString());
jIDni.setText(jtDuenio.getValueAt(filamodificar, 2).toString());
jtnombre.setText(jtDuenio.getValueAt(filamodificar, 3).toString());
jcbEspecie.setSelectedItem(jtDuenio.getValueAt(filamodificar, 4).toString());
jcbRaza.setSelectedItem(jtDuenio.getValueAt(filamodificar, 5).toString());
jtfPeso.setText(jtDuenio.getValueAt(filamodificar, 6).toString());
jtfColor.setText(jtDuenio.getValueAt(filamodificar, 9).toString());
SimpleDateFormat formatoDelTexto = new SimpleDateFormat("yyyy-MM-dd");
String fecha = jtDuenio.getValueAt(jtDuenio.getSelectedRow(), 7).toString().trim();
Date dato = null;

```

```

        try {
            dato = formatoDelTexto.parse(fecha);
        } catch (ParseException ex) {
            ex.printStackTrace();
        }
        jDateChooser1.setDate(dato);
        jcbSexo.setSelectedItem(jtDuenio.getValueAt(filamodificar, 8).toString());
        jlCodAnimal.setText(jtDuenio.getValueAt(filamodificar, 11).toString());
    } else {
        JOptionPane.showMessageDialog(this, "No ha seleccionado ");
    }
}

private void jbBuscarDueñoActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    ListaDueño lista = new ListaDueño();
    PantallaPrincipal.jdpFondo.add(lista);
    listaToFront();
    lista.setVisible(true);
    ListaDueño.jLabel1.setText("2");
}

Object[][] dtAnimal;
MySQL animal = new MySQL();
Object[] dcEspecie;
Object[] dcRaza;
// Variables declaration - do not modify
private com.toedter.calendar.JDateChooser jDateChooser1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPopupMenu jPopupMenu1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JButton jbActualizar;
private javax.swing.JButton jbBuscar;
public static javax.swing.JButton jbBuscarDueño;
private javax.swing.JButton jbCancelar;
private javax.swing.JButton jbNuevaEspecie;
private javax.swing.JButton jbNuevaRaza;
private javax.swing.JComboBox jcbEspecie;
private javax.swing.JComboBox jcbRaza;
private javax.swing.JComboBox jcbSexo;
public static javax.swing.JLabel jlApellido;
private javax.swing.JLabel jlCodAnimal;
public static javax.swing.JLabel jlDni;
public static javax.swing.JLabel jlNombre;
private javax.swing.JMenuItem jmiModificarAnimal;
private javax.swing.JPanel jplImagen;
private javax.swing.JPanel jp_Animal;
private javax.swing.JPanel jp_Dueño;
private javax.swing.JRadioButton jrbDni;
private javax.swing.JRadioButton jrbMascota;
private javax.swing.JTable jtDuenio;
private javax.swing.JTextField jtfColor;
private javax.swing.JTextField jtfPeso;
private javax.swing.JTextField jtfbuscar;
private javax.swing.JTextField jtnombre;
// End of variables declaration
}

```


ModificarAnimalP.Java

```
package ModuloAnimal;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
public class ModificarAnimalP extends javax.swing.JInternalFrame {
    public ModificarAnimalP() {
        initComponents();
        ImgAnimalp img = new ImgAnimalp();
        jplImagen.add(img);
        jplImagen.repaint();
        cargarTabla();
        cargarEspecies();
        cargarRazas();
    }
    private void cargarTabla(){
        String[] columnNames = {"Apellido", "Nombre", "DNI", "Nombre", "Especie", "Raza", "Peso", "Fecha", "Sexo", "Color", "Vivo", "Codigo Animal"};
        String colName[] =
{"apellido", "nombre", "dni_duenio", "animal_pequenos.nombre", "especie", "raza", "peso", "nacimiento", "sexo", "color", "vivo", "cod_animal"};
//nombre de columnas, tabla, instruccion sql
        dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`. * FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
(`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio`) AND (`animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie`) AND
(`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza`)");
// se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtAnimal, columnNames);
        jtDuenio.setModel(datos);
        // se colocan los datos en la tabla
        int a = jtDuenio.getRowCount();
        //depurador de jtDuenio de item null
        int i=0;
        a--;
        while(a>=i){
            if(jtDuenio.getValueAt(i,1)==null){
                ((DefaultTableModel)jtDuenio.getModel()).removeRow(i);
                a--;
                i--;
            }
            i++;
        }
    }
    void desbloquear(){
        jbBuscarDueño.setEnabled(true);
        jtNombre.setEnabled(true);
        jcbEspecie.setEnabled(true);
        jcbRaza.setEnabled(true);
        jbNuevaEspecie.setEnabled(true);
        jbNuevaRaza.setEnabled(true);
        jcbSexo.setEnabled(true);
        jDateChooser1.setEnabled(true);
        jtFPeso.setEnabled(true);
        jtFColor.setEnabled(true);
        jbActualizar.setEnabled(true);
    }
    private void cargarEspecies(){
        //nombre de columnas, tabla, instruccion sql
        dcEspecie = animal.GetColumna("lista_especie", "especie", "SELECT * FROM `lista_especie`");
        // se colocan los datos en la tabla
        DefaultComboBoxModel datos = new DefaultComboBoxModel(dcEspecie);
        jcbEspecie.setModel(datos);
    }
}
```

```

private void cargarRazas(){
    jcbRaza.removeAllItems();
    String especie = jcbEspecie.getSelectedItem().toString();

    String cod_especie = animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especie WHERE especie = \"\"+especie+\"\"");
    //nombre de columnas, tabla, instruccion sql
    dcRaza = animal.GetColumna("lista_raza", "raza", "SELECT * FROM lista_raza WHERE cod_especie = \"\"+cod_especie+\"\"");

    // se colocan los datos en la tabla
    DefaultComboBoxModel datos = new DefaultComboBoxModel(dcRaza);
    jcbRaza.setModel(datos);
    int a = jcbRaza.getItemCount();
    //depurador de jcbRaza de item null
    int i=0;
    a--;
    while(a>=i){
        if(jcbRaza.getItemAt(i)==null){
            jcbRaza.removeItemAt(i);
            a--;
            i--;
        }
        i++;
    }
}

/*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")
private void jcbEspecieItemStateChanged(java.awt.event.ItemEvent evt) {
    cargarRazas();
}

private void jcbEspecieActionPerformed(java.awt.event.ActionEvent evt) {
}

private void jbNuevaEspecieActionPerformed(java.awt.event.ActionEvent evt) {
    String nuevaesp = JOptionPane.showInputDialog(rootPane,"Ingrese la especie que desea agregar:", "Nueva Especie", WIDTH);
    boolean aux = animal.existe("lista_especie", "especie", nuevaesp);
    if (aux==true){
        JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
    }else{
        String datos[] = {nuevaesp};
        //array de datos , instruccion sql
        animal.InsertarRegistro(datos, "insert into lista_especie (`especie`)values(?)");
        cargarEspecies();
        //buscamos el cod_especie correspondiente
        int ultimo= jcbEspecie.getItemCount()-1;
        String especie = jcbEspecie.getItemAt(ultimo).toString();
        String cod_especie = animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especie WHERE especie = \"\"+especie+\"\"");
        //agregamos la nueva raza
        String nuevараza = JOptionPane.showInputDialog(rootPane,"Ingrese una raza inicial para "+nuevaesp+"", "Nueva Raza", WIDTH);
        String dato[] = {cod_especie, nuevараza};
        animal.InsertarRegistro(dato, "insert into lista_raza (`cod_especie`,`raza`)values(?,?)");
        cargarRazas();
    }
}

private void jbNuevaRazaActionPerformed(java.awt.event.ActionEvent evt) {
    String especie= (String)jcbEspecie.getSelectedItem();
    String cod_especie = animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especie WHERE especie = \"\"+especie+\"\"");
    String nuevараza = JOptionPane.showInputDialog(rootPane,"Ingrese una raza para "+especie+"", "Nueva Raza", WIDTH);
    boolean aux = animal.existe("lista_raza", "raza", nuevараza);
    if (aux==true){
        JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
    }else{
        String dato[] = {cod_especie, nuevараza};
        animal.InsertarRegistro(dato, "insert into lista_raza (`cod_especie`,`raza`)values(?,?)");
        cargarRazas();
    }
}

private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}

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

```

```

String nombre = jtNombre.getText();
String especie= (String)jcbEspecie.getSelectedItem();
String cod_especie = animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especie WHERE especie = \"\"+especie+\"\"");
String raza = (String)jcbRaza.getSelectedItem();
String cod_raza = animal.GetData("cod_raza", "SELECT cod_raza FROM lista_raza WHERE raza = \"\"+raza+\"\"");
String peso = jtFPeso.getText();
DateFormat fecha = new SimpleDateFormat("yyyy-MM-dd");
Date nacio = jDateChooser1.getDate();
String date = fecha.format(nacio);
String sexo = (String)jcbSexo.getSelectedItem();
String color= jtFColor.getText();
String dni_duenio = jLDni.getText();
String life ="si";
String cod_animal =jLCodAnimal.getText();
String datos[] = {nombre,cod_especie,cod_raza,peso,date,sexo,color,dni_duenio,life,cod_animal};
//array de datos , instruccion sql
animal.UpdateRegistro(datos, "UPDATE
`animal_pequenos`SET `nombre`=?,`cod_especie`=?,`cod_raza`=?,`peso`=?,`nacimiento`=?,`sexo`=?,`color`=?,`dni_duenio`=?,`vivo`=? WHERE
`cod_animal`=?");
cargarTabla();
}
private void jtFBuscarMouseClicked(java.awt.event.MouseEvent evt) {
}
private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
String texto= jtFBuscar.getText();
String[] columnNames = {"Apellido", "Nombre", "DNI", "Nombre", "Especie", "Raza", "Peso", "Fecha", "Sexo", "Color", "Vivo", "Codigo Animal"};
String colName[] =
{"apellido", "nombre", "dni_duenio", "animal_pequenos.nombre", "especie", "raza", "peso", "nacimiento", "sexo", "color", "vivo", "cod_animal"};
//nombre de columnas, tabla, instruccion sql
//nombre de columnas, tabla, instruccion sql
if(jrbDni.isSelected()==true){
dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`. *FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio` AND `animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie` AND
`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza` AND `duenio`.`dni_duenio` = "+texto+" ");
}
if(jrbMascota.isSelected()==true){
dtAnimal = animal.GetTabla(colName, "animal_pequenos, duenio, lista_especie, lista_raza", "SELECT `duenio`.`nombre`,
`duenio`.`apellido`, `lista_especie`.`especie`, `lista_raza`.`raza`, `animal_pequenos`. *FROM animal_pequenos, duenio, lista_especie, lista_raza WHERE
(`duenio`.`dni_duenio` = `animal_pequenos`.`dni_duenio`) AND (`animal_pequenos`.`cod_especie` = `lista_especie`.`cod_especie`) AND
(`animal_pequenos`.`cod_raza` = `lista_raza`.`cod_raza`) AND (`animal_pequenos`.`nombre` = \"\" + texto + \"\" ");
}
// se colocan los datos en la tabla
DefaultTableModel datos = new DefaultTableModel(dtAnimal, columnNames);
jtDuenio.setModel(datos);
int a = jtDuenio.getRowCount();
//depurador de jtDuenio de item null
int i=0;
a--;
while(a>=i){
if(jtDuenio.getValueAt(i,1)==null){
((DefaultTableModel)jtDuenio.getModel()).removeRow(i);
a--;
i--;
}
i++;
}
}
private void jmiModificarAnimalActionPerformed(java.awt.event.ActionEvent evt) {
desbloquear();
int filamodificar = jtDuenio.getSelectedRow();
if (filamodificar >= 0) {
jApellido.setText(jtDuenio.getValueAt(filamodificar, 0).toString());
jNombre.setText(jtDuenio.getValueAt(filamodificar, 1).toString());
jDni.setText(jtDuenio.getValueAt(filamodificar, 2).toString());
jNombre.setText(jtDuenio.getValueAt(filamodificar, 3).toString());
jcbEspecie.setSelectedItem(jtDuenio.getValueAt(filamodificar, 4).toString());
jcbRaza.setSelectedItem(jtDuenio.getValueAt(filamodificar, 5).toString());
jtfPeso.setText(jtDuenio.getValueAt(filamodificar, 6).toString());
jtfColor.setText(jtDuenio.getValueAt(filamodificar, 9).toString());
SimpleDateFormat formatoDelTexto = new SimpleDateFormat("yyyy-MM-dd");

```

```

String fecha = jtDuenio.getValueAt(jtDuenio.getSelectedRow(),7).toString().trim() ;
Date dato = null;
try {
    dato = formatoDelTexto.parse(fecha);
} catch (ParseException ex) {
    ex.printStackTrace();
}
jDateChooser1.setDate(dato);
jcbSexo.setSelectedItem(jtDuenio.getValueAt(filamodificar, 8).toString());
jlcodAnimal.setText(jtDuenio.getValueAt(filamodificar, 11).toString());
} else {
    JOptionPane.showMessageDialog(this, "No ha seleccionado ");
}
}
private void jbBuscarDueñoActionPerformed(java.awt.event.ActionEvent evt) {
    ListaDueño lista = new ListaDueño();
    PantallaPrincipal.jdpFondo.add(lista);
    listaToFront();
    lista.setVisible(true);
    ListaDueño.jLabel1.setText("2");
}
Object[][] dtAnimal;
MySQL animal = new MySQL();
Object[] dcEspecie;
Object[] dcRaza;
// Variables declaration - do not modify
private com.toedter.calendar.JDateChooser jDateChooser1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPopupMenu jPopupMenu1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JButton jbActualizar;
private javax.swing.JButton jbBuscar;
public static javax.swing.JButton jbBuscarDueño;
private javax.swing.JButton jbCancelar;
private javax.swing.JButton jbNuevaEspecie;
private javax.swing.JButton jbNuevaRaza;
private javax.swing.JComboBox jcbEspecie;
private javax.swing.JComboBox jcbRaza;
private javax.swing.JComboBox jcbSexo;
public static javax.swing.JLabel jlApellido;
private javax.swing.JLabel jlCodAnimal;
public static javax.swing.JLabel jlDni;
public static javax.swing.JLabel jlNombre;
private javax.swing.JMenuItem jmiModificarAnimal;
private javax.swing.JPanel jplmagen;
private javax.swing.JPanel jp_Animal;
private javax.swing.JPanel jp_Dueño;
private javax.swing.JRadioButton jrbDni;
private javax.swing.JRadioButton jrbMascota;
private javax.swing.JTable jtDuenio;
private javax.swing.JTextField jtfColor;
private javax.swing.JTextField jtfPeso;
private javax.swing.JTextField jtfbuscar;
private javax.swing.JTextField jtnombre;
// End of variables declaration
}

```

NuevoAnimalG.Java

```

package ModuloAnimal;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import java.sql.SQLException;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JOptionPane;
public class NuevoAnimalG extends javax.swing.JFrame {
    public NuevoAnimalG() {
        initComponents();
        ImgAnimalp img = new ImgAnimalp();
        jpimagen.add(img);
        jpimagen.repaint();
        cargarEspecies();
        cargarRazas();
    }
    private void cargarEspecies(){
        //nombre de columnas, tabla, instruccion sql
        dcEspecie = animal.GetColumna("lista_especieg", "especie", "SELECT * FROM `lista_especieg`");
        // se colocan los datos en la tabla
        DefaultComboBoxModel datos = new DefaultComboBoxModel(dcEspecie);
        jcbEspecie.setModel(datos);
    }
    private void cargarRazas(){
        jcbRaza.removeAllItems();
        String especie = jcbEspecie.getSelectedItem().toString();
        String cod_especie = animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especieg WHERE especie = \"\""+especie+"\"");
        //nombre de columnas, tabla, instruccion sql
        dcRaza = animal.GetColumna("lista_razag", "raza", "SELECT * FROM lista_razag WHERE cod_especie = \"\""+cod_especie+"\"");
        // se colocan los datos en la tabla
        DefaultComboBoxModel datos = new DefaultComboBoxModel(dcRaza);
        jcbRaza.setModel(datos);
        int a = jcbRaza.getItemCount();
        //depurador de jcbRaza de item null
        int i=0;
        a--;
        while(a>=i){
            if(jcbRaza.getItemAt(i)==null){
                jcbRaza.removeItemAt(i);
                a--;
                i--;
            }
            i++;
        }
    }
    /*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")
    private void jbBuscarDueñoActionPerformed(java.awt.event.ActionEvent evt) {
        ListaDueñoG lista = new ListaDueñoG();
        PantallaPrincipal.jdpFondo.add(lista);
        listaToFront();
        lista.setVisible(true);
        ListaDueñoG.jLabel1.setText("1");
    }
    private void jcbEspecieItemStateChanged(java.awt.event.ItemEvent evt) {
        cargarRazas();
    }
    private void jbNuevaEspecieActionPerformed(java.awt.event.ActionEvent evt) {
        String nuevaesp = JOptionPane.showInputDialog(rootPane,"Ingrese la especie que desea agregar:","Nueva Especie", WIDTH);
        boolean aux = animal.existe("lista_especieg", "especie", nuevaesp);
        if (aux==true){
            JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
        }
    }
}

```

```

    }else{
        String datos[] = {nuevaesp};
        //array de datos , instruccion sql
        animal.InsertarRegistro(datos, "insert into lista_especieg (`especie`)values(?)");
        cargarEspecies();
        //buscamos el cod_especie correspondiente
        int ultimo= jcbEspecie.getItemCount()-1;
        String especie = jcbEspecie.getItemAt(ultimo).toString();
        String cod_especie = animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especieg WHERE especie = \"'+especie+'\"");
//agregamos la nueva raza
        String nuevараза = JOptionPane.showInputDialog(rootPane, "Ingrese una raza inicial para "+nuevaesp+"", "Nueva Raza", WIDTH);
        String dato[] = {cod_especie, nuevараза};
        animal.InsertarRegistro(dato, "insert into lista_razag (`cod_especie`, `raza`)values(?,?)");
        cargarRazas();
    }
}

private void jcbEspecieActionPerformed(java.awt.event.ActionEvent evt) {
}

private void jbNuevaRazaActionPerformed(java.awt.event.ActionEvent evt) {
    String especie= (String)jcbEspecie.getSelectedItem();
    String cod_especie = animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especieg WHERE especie = \"'+especie+'\"");
    String nuevараза = JOptionPane.showInputDialog(rootPane, "Ingrese una raza para "+especie+"", "Nueva Raza", WIDTH);
    boolean aux = animal.existe("lista_razag", "raza", nuevараза);
    if (aux==true){
        JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
    }else{
        String dato[] = {cod_especie, nuevараза};
        animal.InsertarRegistro(dato, "insert into lista_razag (`cod_especie`, `raza`)values(?,?)");
        cargarRazas();
    }
}

private void jbAgregarActionPerformed(java.awt.event.ActionEvent evt) {
    String nombre = jtnombre.getText();
    String especie= (String)jcbEspecie.getSelectedItem();
    //int cod_especie = Integer.parseInt(animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especie WHERE especie =
    \"'+especie+'\""));
    String cod_especie = animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especieg WHERE especie = \"'+especie+'\"");
    String raza = (String)jcbRaza.getSelectedItem();
    //int cod_raza = Integer.parseInt(animal.GetData("cod_raza", "SELECT cod_raza FROM lista_raza WHERE raza = \"'+raza+'\""));
    String cod_raza = animal.GetData("cod_raza", "SELECT cod_raza FROM lista_razag WHERE raza = \"'+raza+'\"");
    //int peso = Integer.parseInt(jtfPeso.getText());
    String peso = jtfPeso.getText();
    //Date nacio = jDateChooser1.getDate();
    DateFormat fecha = new SimpleDateFormat("yyyy-MM-dd");
    Date nacio = jDateChooser1.getDate();
    String date = fecha.format(nacio);
    String sexo = (String)jcbSexo.getSelectedItem();
    String color= jtfColor.getText();
    //int dni = Integer.parseInt(jlDni.getText());
    String dni = jlDni.getText();
    String life ="si";
    boolean aux = animal.existe("animal_grandes", "nombre", nombre);
    if (aux==true){
        JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
    }else{
        String datos[] = {nombre, cod_especie, cod_raza, peso, date, sexo, color, dni, life};
        //array de datos , instruccion sql
        animal.InsertarRegistro(datos, "insert into `animal_grandes` ( `nombre`, `cod_especie`, `cod_raza`, `peso`, `nacimiento`, `sexo`, `color`,
        `dni_duenio`, `vivo`)values(?,?,?,?,?,?,?,?)");
    }
}

private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}

MySQL animal = new MySQL();
Object[] dcEspecie;
Object[] dcRaza;
// Variables declaration - do not modify
private com.toedter.calendar.JDateChooser jDateChooser1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;

```



```
private javax.swing.JLabel jLabel2;  
private javax.swing.JLabel jLabel3;  
private javax.swing.JLabel jLabel4;  
private javax.swing.JLabel jLabel5;  
private javax.swing.JLabel jLabel6;  
private javax.swing.JLabel jLabel7;  
private javax.swing.JLabel jLabel8;  
private javax.swing.JLabel jLabel9;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JButton jbAgrega;  
public static javax.swing.JButton jbBuscarDueño;  
private javax.swing.JButton jbCancelar;  
private javax.swing.JButton jbNuevaEspecie;  
private javax.swing.JButton jbNuevaRaza;  
private javax.swing.JComboBox jcbEspecie;  
private javax.swing.JComboBox jcbRaza;  
private javax.swing.JComboBox jcbSexo;  
public static javax.swing.JLabel jlApellido;  
public static javax.swing.JLabel jlDni;  
public static javax.swing.JLabel jlNombre;  
private javax.swing.JPanel jpAnimal;  
private javax.swing.JPanel jpDueño;  
private javax.swing.JPanel jpImagen;  
private javax.swing.JTextField jtfColor;  
private javax.swing.JTextField jtfPeso;  
private javax.swing.JTextField jtnombre;  
// End of variables declaration  
}
```


NuevoAnimalP.Java

```
package ModuloAnimal;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import java.sql.SQLException;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JOptionPane;
public class NuevoAnimalP extends javax.swing.JInternalFrame {
    public NuevoAnimalP() {
        initComponents();
        ImgAnimalp img = new ImgAnimalp();
        jpimagen.add(img);
        jpimagen.repaint();
        cargarEspecies();
        cargarRazas();
    }
    private void cargarEspecies(){
        //nombre de columnas, tabla, instruccion sql
        dcEspecie = animal.GetColumna("lista_especie", "especie", "SELECT * FROM `lista_especie`");
        // se colocan los datos en la tabla
        DefaultComboBoxModel datos = new DefaultComboBoxModel(dcEspecie);
        jcbEspecie.setModel(datos);
    }
    private void cargarRazas(){
        jcbRaza.removeAllItems();
        String especie = jcbEspecie.getSelectedItem().toString();
        String cod_especie = animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especie WHERE especie = \"\""+especie+"\"");
        //nombre de columnas, tabla, instruccion sql
        dcRaza = animal.GetColumna("lista_raza", "raza", "SELECT * FROM lista_raza WHERE cod_especie = \"\""+cod_especie+"\"");
        // se colocan los datos en la tabla
        DefaultComboBoxModel datos = new DefaultComboBoxModel(dcRaza);
        jcbRaza.setModel(datos);
        int a = jcbRaza.getItemCount();
        //depurador de jcbRaza de item null
        int i=0;
        a--;
        while(a>=i){
            if(jcbRaza.getItemAt(i)==null){
                jcbRaza.removeItemAt(i);
                a--;
                i--;
            }
            i++;
        }
    }
    /*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")
    private void jbBuscarDueñoActionPerformed(java.awt.event.ActionEvent evt) {
        ListaDueño lista = new ListaDueño();
        PantallaPrincipal.jdpFondo.add(lista);
        listaToFront();
        lista.setVisible(true);
        ListaDueño.jLabel1.setText("1");
    }
    private void jcbEspecieItemStateChanged(java.awt.event.ItemEvent evt) {
        cargarRazas();
    }
    private void jbNuevaEspecieActionPerformed(java.awt.event.ActionEvent evt) {
        String nuevaesp = JOptionPane.showInputDialog(rootPane,"Ingrese la especie que desea agregar:", "Nueva Especie", WIDTH);
        boolean aux = animal.existe("lista_especie", "especie", nuevaesp);
        if (aux==true){
            JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
        }
    }
}
```

```

    }else{
        String datos[] = {nuevaesp};
//array de datos , instruccion sql
        animal.InsertarRegistro(datos, "insert into lista_especie (`especie`)values(?)");
        cargarEspecies();
//buscamos el cod_especie correspondiente
        int ultimo= jcbEspecie.getItemCount()-1;
        String especie = jcbEspecie.getItemAt(ultimo).toString();
        String cod_especie = animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especie WHERE especie = \""+especie+"\"");
//agregamos la nueva raza
        String nuevараза = JOptionPane.showInputDialog(rootPane,"Ingrese una raza inicial para "+nuevaesp+"", "Nueva Raza",WIDTH);
        String dato[]= {cod_especie,nuevaraza};
        animal.InsertarRegistro(dato, "insert into lista_raza (`cod_especie`,`raza`)values(?,?)");
        cargarRazas();
    }
}
private void jcbEspecieActionPerformed(java.awt.event.ActionEvent evt) {
}
private void jbNuevaRazaActionPerformed(java.awt.event.ActionEvent evt) {
    String especie= (String)jcbEspecie.getSelectedItem();
    String cod_especie = animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especie WHERE especie = \""+especie+"\"");
    String nuevараза = JOptionPane.showInputDialog(rootPane,"Ingrese una raza para "+especie+"", "Nueva Raza",WIDTH);
    boolean aux = animal.existe("lista_raza", "raza", nuevараза);
    if (aux==true){
        JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
    }else{
        String dato[]= {cod_especie,nuevaraza};
        animal.InsertarRegistro(dato, "insert into lista_raza (`cod_especie`,`raza`)values(?,?)");
        cargarRazas();
    }
}
private void jbAgregarActionPerformed(java.awt.event.ActionEvent evt) {
    String nombre = jtnombre.getText();
    String especie= (String)jcbEspecie.getSelectedItem();
    //int cod_especie = Integer.parseInt(animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especie WHERE especie =
    \""+especie+"\""));
    String cod_especie =animal.GetData("cod_especie", "SELECT cod_especie FROM lista_especie WHERE especie = \""+especie+"\"");
    String raza = (String)jcbRaza.getSelectedItem();
    //int cod_raza = Integer.parseInt(animal.GetData("cod_raza", "SELECT cod_raza FROM lista_raza WHERE raza = \""+raza+"\""));
    String cod_raza = animal.GetData("cod_raza", "SELECT cod_raza FROM lista_raza WHERE raza = \""+raza+"\"");
    //int peso = Integer.parseInt(jtfPeso.getText());
    String peso = jtfPeso.getText();
    //Date nacio = jDateChooser1.getDate();
    DateFormat fecha = new SimpleDateFormat("yyyy-MM-dd");
    Date nacio = jDateChooser1.getDate();
    String date = fecha.format(nacio);
    String sexo = (String)jcbSexo.getSelectedItem();
    String color= jtfColor.getText();
    //int dni = Integer.parseInt(jlDni.getText());
    String dni = jlDni.getText();
    String life ="si";
    boolean aux = animal.existe("animal_pequenos", "nombre", nombre);
    if (aux==true){
        JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
    }else{
        String datos[] = {nombre,cod_especie,cod_raza,peso,date,sexo,color,dni,life};
        //array de datos , instruccion sql
        animal.InsertarRegistro(datos, "insert into `animal_pequenos`(` nombre`, `cod_especie`, `cod_raza`, `peso`, `nacimiento`, `sexo`, `color`,
        `dni_duenio`, `vivo`)values(?,?,?,?,?,?,?,?)");
    }
}
private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}
MySQL animal = new MySQL();
Object[] dcEspecie;
Object[] dcRaza;
// Variables declaration - do not modify
private com.toedter.calendar.JDateChooser jDateChooser1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;

```

```
private javax.swing.JLabel jLabel2;  
private javax.swing.JLabel jLabel3;  
private javax.swing.JLabel jLabel4;  
private javax.swing.JLabel jLabel5;  
private javax.swing.JLabel jLabel6;  
private javax.swing.JLabel jLabel7;  
private javax.swing.JLabel jLabel8;  
private javax.swing.JLabel jLabel9;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JButton jbAgrega;  
public static javax.swing.JButton jbBuscarDueño;  
private javax.swing.JButton jbCancelar;  
private javax.swing.JButton jbNuevaEspecie;  
private javax.swing.JButton jbNuevaRaza;  
private javax.swing.JComboBox jcbEspecie;  
private javax.swing.JComboBox jcbRaza;  
private javax.swing.JComboBox jcbSexo;  
public static javax.swing.JLabel jlApellido;  
public static javax.swing.JLabel jlDni;  
public static javax.swing.JLabel jlNombre;  
private javax.swing.JPanel jpAnimal;  
private javax.swing.JPanel jpDueño;  
private javax.swing.JPanel jpImagen;  
private javax.swing.JTextField jtfColor;  
private javax.swing.JTextField jtfPeso;  
private javax.swing.JTextField jtnombre;  
// End of variables declaration  
}
```

Paquete ModuloDueño

ImgDuenio.Java

```
package ModuloDueño;
import java.awt.Graphics;
import javax.swing.ImageIcon;
public class ImgDuenio extends javax.swing.JPanel {
    public ImgDuenio(){
        this.setSize(128,128);
    }
    @Override
    public void paint(Graphics g){
        ImageIcon imagenFondo = new ImageIcon(getClass().getResource("/imagenes/duenio.png"));
        g.drawImage(imagenFondo.getImage(),0,0,128,128, null);
        setOpaque(false);
        super.paintComponent(g);
    }
}
```

ModificarDueño.Java

```
package ModuloDueño;
import Clases.MySQL;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
public class ModificarDueño extends javax.swing.JInternalFrame {
    public ModificarDueño() {
        initComponents();
        ImgDuenio img = new ImgDuenio();
        jplImagen.add(img);
        jplImagen.repaint();
        cargarTabla();
    }
    private void cargarTabla() {
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Facebook", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_uenio", "email", "facebook", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        dtDuenio = duenio.GetTabla(colName, "duenio", "SELECT * FROM `duenio` ORDER BY `apellido`");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtDuenio, columnNames);
        jtDuenio.setModel(datos);
    }
    void desbloquear() {
        jtfNombre.setEnabled(true);
        jtfApellido.setEnabled(true);
        jtfEmail.setEnabled(true);
        jtfFijo.setEnabled(true);
        jtfCelular.setEnabled(true);
        jtfDomicilio.setEnabled(true);
        jtfFacebook.setEnabled(true);
        jtfCiudad.setEnabled(true);
        jbActualizar.setEnabled(true);
    }
    /*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")
    private void jbActualizarActionPerformed(java.awt.event.ActionEvent evt) {
        String dni_uenio = jtfDNI.getText();
        String nombre = jtfNombre.getText();
        String apellido = jtfApellido.getText();
        String email = jtfEmail.getText();
        String facebook = jtfFacebook.getText();
        String domicilio = jtfDomicilio.getText();
        String ciudad = jtfCiudad.getText();
        String telefono_fijo = jtfFijo.getText();
        String telefono_celular = jtfCelular.getText();
        String datos[] = {nombre, apellido, email, facebook, domicilio, ciudad, telefono_fijo, telefono_celular, dni_uenio};
        //array de datos , instruccion sql
        duenio.UpdateRegistro(datos, "UPDATE `duenio` SET
`nombre`=?,`apellido`=?,`email`=?,`facebook`=?,`domicilio`=?,`ciudad`=?,`telefono_fijo`=?,`telefono_celular`=? WHERE `dni_uenio`=?");
        cargarTabla();
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    private void jmiModificarActionPerformed(java.awt.event.ActionEvent evt) {
        desbloquear();
        int filamodificar = jtDuenio.getSelectedRow();
        if (filamodificar >= 0) {
            jtfApellido.setText(jtDuenio.getValueAt(filamodificar, 0).toString());
            jtfNombre.setText(jtDuenio.getValueAt(filamodificar, 1).toString());
            jtfDNI.setText(jtDuenio.getValueAt(filamodificar, 2).toString());
            jtfEmail.setText(jtDuenio.getValueAt(filamodificar, 3).toString());
            jtfFacebook.setText(jtDuenio.getValueAt(filamodificar, 4).toString());
            jtfDomicilio.setText(jtDuenio.getValueAt(filamodificar, 5).toString());
            jtfCiudad.setText(jtDuenio.getValueAt(filamodificar, 6).toString());
        }
    }
}
```

```

        jtfCelular.setText(jtDuenio.getValueAt(filamodificar, 7).toString());
        jtFfijo.setText(jtDuenio.getValueAt(filamodificar, 8).toString());
    } else {
        JOptionPane.showMessageDialog(this, "No ha seleccionado ");
    }
}
private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
    String texto = jtfbuscar.getText();
    String[] columNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Facebook", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
    String colName[] = {"apellido", "nombre", "dni_duenio", "email", "facebook", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
    //nombre de columnas, tabla, instruccion sql
    if (jrbDni.isSelected() == true) {
        dtDuenio = duenio.GetTabla(colName, "duenio", "SELECT * FROM `duenio` WHERE `dni_duenio`=" + texto + " ");
    }
    if (jrbApellido.isSelected() == true) {
        dtDuenio = duenio.GetTabla(colName, "duenio", "SELECT * FROM `duenio` WHERE `apellido` = \"" + texto + "\"");
    }
    // se colocan los datos en la tabla
    DefaultTableModel datos = new DefaultTableModel(dtDuenio, columNames);
    jtDuenio.setModel(datos);
}
MySQL duenio = new MySQL();
Object[][] dtDuenio;
int fila = -1;
// Variables declaration - do not modify
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPopupMenu jPopupMenu1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JButton jbActualizar;
private javax.swing.JButton jbBuscar;
private javax.swing.JButton jbCancelar;
private javax.swing.JMenuItem jmiModificar;
private javax.swing.JPanel jPanelImagen;
private javax.swing.JPanel jp_duenio;
private javax.swing.JRadioButton jrbApellido;
private javax.swing.JRadioButton jrbDni;
private javax.swing.JTable jtDuenio;
private javax.swing.JTextField jtfApellido;
private javax.swing.JTextField jtfCelular;
private javax.swing.JTextField jtfCiudad;
private javax.swing.JTextField jtfDNI;
private javax.swing.JTextField jtfDomicilio;
private javax.swing.JTextField jtfEmail;
private javax.swing.JTextField jtfFacebook;
private javax.swing.JTextField jtfFijo;
private javax.swing.JTextField jtfNombre;
private javax.swing.JTextField jtfbuscar; // End of variables declaration }

```

NuevoDueño.Java

```
package ModuloDueño;
import Clases.MySQL;
import javax.swing.JOptionPane;
public class NuevoDueño extends javax.swing.JFrame {
    public NuevoDueño() {
        initComponents();
        ImgDuenio img = new ImgDuenio ();
        jplmagen.add(img);
        jplmagen.repaint();}
    /*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")
    private void jbAgregarActionPerformed(java.awt.event.ActionEvent evt) {
        String dni = jtfDNI.getText();
        boolean aux = duenio.existe("duenio", "dni_duenio", dni);
        if (aux==true){
            JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
        }else{
            String dni_duenio = jtfDNI.getText();
            String nombre = jtfNombre.getText();
            String apellido= jtfApellido.getText();
            String email = jtfEmail.getText();
            String facebook = jtfFacebook.getText();
            String domicilio = jtfDomicilio.getText();
            String ciudad = jtfCiudad.getText();
            String telefono_fijo = jtfFijo.getText();
            String telefono_celular = jtfCelular.getText();
            String datos[] = {dni_duenio,nombre,apellido,email,facebook,domicilio,ciudad,telefono_fijo,telefono_celular};
            //array de datos , instruccion sql
            duenio.InsertarRegistro(datos, "insert into duenio values(?,?,?,?,?,?,?,?)");    }
        }
        private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {    this.dispose();}
        MySQL duenio = new MySQL();
        Object[][] dtDuenio;
        int fila = -1;
        // Variables declaration - do not modify
        private javax.swing.JLabel jLabel1;
        private javax.swing.JLabel jLabel10;
        private javax.swing.JLabel jLabel2;
        private javax.swing.JLabel jLabel3;
        private javax.swing.JLabel jLabel4;
        private javax.swing.JLabel jLabel5;
        private javax.swing.JLabel jLabel6;
        private javax.swing.JLabel jLabel7;
        private javax.swing.JLabel jLabel8;
        private javax.swing.JLabel jLabel9;
        private javax.swing.JPanel jPanel1;
        private javax.swing.JButton jbAgregar;
        private javax.swing.JButton jbCancelar;
        private javax.swing.JPanel jplmagen;
        private javax.swing.JPanel jp_duenio;
        private javax.swing.JTextField jtfApellido;
        private javax.swing.JTextField jtfCelular;
        private javax.swing.JTextField jtfCiudad;
        private javax.swing.JTextField jtfDNI;
        private javax.swing.JTextField jtfDomicilio;
        private javax.swing.JTextField jtfEmail;
        private javax.swing.JTextField jtfFacebook;
        private javax.swing.JTextField jtfFijo;
        private javax.swing.JTextField jtfNombre;// End of variables declaration    }
```


Paquetes ModuloUsuario

ImgDoctor.Java

```
package ModuloUsuario;
import java.awt.Graphics;
import javax.swing.ImageIcon;
public class ImgDoctor extends javax.swing.JPanel {
    public ImgDoctor(){
        this.setSize(128,128);
    }
    @Override
    public void paint(Graphics g){
        ImageIcon imagenFondo = new ImageIcon(getClass().getResource("/imagenes/duenio.png"));
        g.drawImage(imagenFondo.getImage(),0,0,128,128, null);
        setOpaque(false);
        super.paintComponent(g);
    }
}
```

ModificarAlumno.Java

```
package ModuloUsuario;
import Clases.MySQL;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
public class ModificarAlumno extends javax.swing.JInternalFrame {
    public ModificarAlumno() {
        initComponents();
        ImgDoctor img = new ImgDoctor();
        jplImagen.add(img);
        jplImagen.repaint();
        cargarTabla(); }
    private void cargarTabla() {
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_alumno", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        dtAlumno = alumno.GetTabla(colName, "alumno", "SELECT * FROM `alumno` ORDER BY `apellido`");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtAlumno, columnNames);
        jtAlumno.setModel(datos);
    }
    void desbloquear() {
        jtfNombre.setEnabled(true);
        jtfApellido.setEnabled(true);
        jtfEmail.setEnabled(true);
        jtfFijo.setEnabled(true);
        jtfCelular.setEnabled(true);
        jtfDomicilio.setEnabled(true);
        jtfMatricula.setEnabled(true);
        jtfCiudad.setEnabled(true);
        jbActualizar.setEnabled(true);
    }
    /*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")
    private void jbActualizarActionPerformed(java.awt.event.ActionEvent evt) {
        String dni_uenio = jtfDNI.getText();
        String nombre = jtfNombre.getText();
        String apellido = jtfApellido.getText();
        String email = jtfEmail.getText();
        String facebook = jtfMatricula.getText();
        String domicilio = jtfDomicilio.getText();
        String ciudad = jtfCiudad.getText();
        String telefono_fijo = jtfFijo.getText();
        String telefono_celular = jtfCelular.getText();
        String datos[] = {nombre, apellido, email, facebook, domicilio, ciudad, telefono_fijo, telefono_celular, dni_uenio};
        //array de datos , instruccion sql
        alumno.UpdateRegistro(datos, "UPDATE `alumno` SET
`nombre`=?,`apellido`=?,`email`=?,`matricula`=?,`domicilio`=?,`ciudad`=?,`telefono_fijo`=?,`telefono_celular`=? WHERE `dni_alumno`=?");
        cargarTabla();
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose(); }
    private void jmiModificarActionPerformed(java.awt.event.ActionEvent evt) {
        desbloquear();
        int filamodificar = jtAlumno.getSelectedRow();
        if (filamodificar >= 0) {
            jtfApellido.setText(jtAlumno.getValueAt(filamodificar, 0).toString());
            jtfNombre.setText(jtAlumno.getValueAt(filamodificar, 1).toString());
            jtfDNI.setText(jtAlumno.getValueAt(filamodificar, 2).toString());
            jtfEmail.setText(jtAlumno.getValueAt(filamodificar, 3).toString());
            jtfMatricula.setText(jtAlumno.getValueAt(filamodificar, 4).toString());
            jtfDomicilio.setText(jtAlumno.getValueAt(filamodificar, 5).toString());
            jtfCiudad.setText(jtAlumno.getValueAt(filamodificar, 6).toString());
            jtfCelular.setText(jtAlumno.getValueAt(filamodificar, 7).toString());
            jtfFijo.setText(jtAlumno.getValueAt(filamodificar, 8).toString());
        }
    }
}
```

```

    } else {
        JOptionPane.showMessageDialog(this, "No ha seleccionado ");
    }
}

private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
    String texto = jtfbuscar.getText();
    String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
    String colName[] = {"apellido", "nombre", "dni_alumno", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
    //nombre de columnas, tabla, instruccion sql
    if (jrbDni.isSelected() == true) {
        dtAlumno = alumno.GetTabla(colName, "alumno", "SELECT * FROM `duenio` WHERE `dni_alumno`=" + texto + " ");
    }
    if (jrbApellido.isSelected() == true) {
        dtAlumno = alumno.GetTabla(colName, "alumno", "SELECT * FROM `doctor` WHERE `apellido` = '\"" + texto + "\"");
    }
    // se colocan los datos en la tabla
    DefaultTableModel datos = new DefaultTableModel(dtAlumno, columnNames);
    jtAlumno.setModel(datos);
}

MySQL alumno = new MySQL();
Object[][] dtAlumno;
int fila = -1;
// Variables declaration - do not modify
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPopupMenu jPopupMenu1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JButton jbActualizar;
private javax.swing.JButton jbBuscar;
private javax.swing.JButton jbCancelar;
private javax.swing.JMenuItem jmiModificar;
private javax.swing.JPanel jPanelImagen;
private javax.swing.JPanel jp_duenio;
private javax.swing.JRadioButton jrbApellido;
private javax.swing.JRadioButton jrbDni;
private javax.swing.JTable jtAlumno;
private javax.swing.JTextField jtfApellido;
private javax.swing.JTextField jtfCelular;
private javax.swing.JTextField jtfCiudad;
private javax.swing.JTextField jtfDNI;
private javax.swing.JTextField jtfDomicilio;
private javax.swing.JTextField jtfEmail;
private javax.swing.JTextField jtfFijo;
private javax.swing.JTextField jtfMatricula;
private javax.swing.JTextField jtfNombre;
private javax.swing.JTextField jtfbuscar;
// End of variables declaration    }

```

ModificarDoctor.Java

```
package ModuloUsuario;
import Clases.MySQL;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
public class ModificarDoctor extends javax.swing.JInternalFrame {
    public ModificarDoctor() {
        initComponents();
        ImgDoctor img = new ImgDoctor();
        jplImagen.add(img);
        jplImagen.repaint();
        cargarTabla();
    }
    private void cargarTabla() {
        String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
        String colName[] = {"apellido", "nombre", "dni_doctor", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
        //nombre de columnas, tabla, instruccion sql
        dtDoctor = doctor.GetTabla(colName, "doctor", "SELECT * FROM `doctor` ORDER BY `apellido`");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtDoctor, columnNames);
        jtDoctor.setModel(datos); }
    void desbloquear() {
        jtfNombre.setEnabled(true);
        jtfApellido.setEnabled(true);
        jtfEmail.setEnabled(true);
        jtfFijo.setEnabled(true);
        jtfCelular.setEnabled(true);
        jtfDomicilio.setEnabled(true);
        jtfMatricula.setEnabled(true);
        jtfCiudad.setEnabled(true);
        jbActualizar.setEnabled(true); }
    /*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")
    private void jbActualizarActionPerformed(java.awt.event.ActionEvent evt) {
        String dni_duenio = jtfDNI.getText();
        String nombre = jtfNombre.getText();
        String apellido = jtfApellido.getText();
        String email = jtfEmail.getText();
        String facebook = jtfMatricula.getText();
        String domicilio = jtfDomicilio.getText();
        String ciudad = jtfCiudad.getText();
        String telefono_fijo = jtfFijo.getText();
        String telefono_celular = jtfCelular.getText();
        String datos[] = {nombre, apellido, email, facebook, domicilio, ciudad, telefono_fijo, telefono_celular, dni_duenio};
        //array de datos , instruccion sql
        doctor.UpdateRegistro(datos, "UPDATE `doctor` SET
`nombre`=?,`apellido`=?,`email`=?,`matricula`=?,`domicilio`=?,`ciudad`=?,`telefono_fijo`=?,`telefono_celular`=? WHERE `dni_doctor`=?");
        cargarTabla();
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    private void jmiModificarActionPerformed(java.awt.event.ActionEvent evt) {
        desbloquear();
        int filamodificar = jtDoctor.getSelectedRow();
        if (filamodificar >= 0) {
            jtfApellido.setText(jtDoctor.getValueAt(filamodificar, 0).toString());
            jtfNombre.setText(jtDoctor.getValueAt(filamodificar, 1).toString());
            jtfDNI.setText(jtDoctor.getValueAt(filamodificar, 2).toString());
            jtfEmail.setText(jtDoctor.getValueAt(filamodificar, 3).toString());
            jtfMatricula.setText(jtDoctor.getValueAt(filamodificar, 4).toString());
            jtfDomicilio.setText(jtDoctor.getValueAt(filamodificar, 5).toString());
            jtfCiudad.setText(jtDoctor.getValueAt(filamodificar, 6).toString());
            jtfCelular.setText(jtDoctor.getValueAt(filamodificar, 7).toString());
            jtfFijo.setText(jtDoctor.getValueAt(filamodificar, 8).toString());
        }
    }
}
```

```

    } else {
        JOptionPane.showMessageDialog(this, "No ha seleccionado ");
    }
}

private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
    String texto = jtfbuscar.getText();
    String[] columnNames = {"Apellido", "Nombre", "DNI", "E-Mail", "Matricula", "Domicilio", "Ciudad", "Celular", "Tel.Fijo"};
    String colName[] = {"apellido", "nombre", "dni_doctor", "email", "matricula", "domicilio", "ciudad", "telefono_celular", "telefono_fijo"};
    //nombre de columnas, tabla, instruccion sql
    if (jrbDni.isSelected() == true) {
        dtDoctor = doctor.GetTabla(colName, "doctor", "SELECT * FROM `duenio` WHERE `dni_doctor`='" + texto + "'");
    }
    if (jrbApellido.isSelected() == true) {
        dtDoctor = doctor.GetTabla(colName, "doctor", "SELECT * FROM `doctor` WHERE `apellido` = '\" + texto + '\"");
    }
}

// se colocan los datos en la tabla
DefaultTableModel datos = new DefaultTableModel(dtDoctor, columnNames);
jtDoctor.setModel(datos);
}

MySQL doctor = new MySQL();
Object[][] dtDoctor;
int fila = -1;
// Variables declaration - do not modify
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPopupMenu jPopupMenu1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JButton jbActualizar;
private javax.swing.JButton jbBuscar;
private javax.swing.JButton jbCancelar;
private javax.swing.JMenuItem jmiModificar;
private javax.swing.JPanel jPanelImagen;
private javax.swing.JPanel jp_duenio;
private javax.swing.JRadioButton jrbApellido;
private javax.swing.JRadioButton jrbDni;
private javax.swing.JTable jtDoctor;
private javax.swing.JTextField jtfApellido;
private javax.swing.JTextField jtfCelular;
private javax.swing.JTextField jtfCiudad;
private javax.swing.JTextField jtfDNI;
private javax.swing.JTextField jtfDomicilio;
private javax.swing.JTextField jtfEmail;
private javax.swing.JTextField jtfFijo;
private javax.swing.JTextField jtfMatricula;
private javax.swing.JTextField jtfNombre;
private javax.swing.JTextField jtfbuscar;
// End of variables declaration }

```

NuevoAlumno.Java

```
package ModuloUsuario;
import Clases.MySQL;
import javax.swing.JOptionPane;
public class NuevoAlumno extends javax.swing.JInternalFrame {
    public NuevoAlumno() {
        initComponents();
        ImgDoctor img = new ImgDoctor ();
        jplmagen.add(img);
        jplmagen.repaint();}
    /*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")
    private void jbAgregarActionPerformed(java.awt.event.ActionEvent evt) {
        String dni = jtfDNI.getText();
        boolean aux = alumno.existe("alumno", "dni_alumno", dni);
        if (aux==true){
            JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
        }else{
            String dni_alumno = jtfDNI.getText();
            String nombre = jtfNombre.getText();
            String apellido= jtfApellido.getText();
            String email = jtfEmail.getText();
            String matricula = jtfMatricula.getText();
            String domicilio = jtfDomicilio.getText();
            String ciudad = jtfCiudad.getText();
            String telefono_fijo = jtfFijo.getText();
            String telefono_celular = jtfCelular.getText();
            String datos[] = {dni_alumno,nombre,apellido,email,matricula,domicilio,ciudad,telefono_fijo,telefono_celular};
            //array de datos , instruccion sql
            alumno.InsertarRegistro(datos, "insert into alumno values(?,?,?,?,?,?,?,?,?)"); }
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {      this.dispose(); }
    MySQL alumno = new MySQL();
    Object[][] dtAlumno;
    int fila = -1;
    // Variables declaration - do not modify
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JButton jbAgregar;
    private javax.swing.JButton jbCancelar;
    private javax.swing.JPanel jplmagen;
    private javax.swing.JPanel jp_duenio;
    private javax.swing.JTextField jtfApellido;
    private javax.swing.JTextField jtfCelular;
    private javax.swing.JTextField jtfCiudad;
    private javax.swing.JTextField jtfDNI;
    private javax.swing.JTextField jtfDomicilio;
    private javax.swing.JTextField jtfEmail;
    private javax.swing.JTextField jtfFijo;
    private javax.swing.JTextField jtfMatricula;
    private javax.swing.JTextField jtfNombre; // End of variables declaration }
```

NuevoDoctor.Java

```
package ModuloUsuario;
import Clases.MySQL;
import javax.swing.JOptionPane;
public class NuevoDoctor extends javax.swing.JFrame {
    public NuevoDoctor() {
        initComponents();
        ImgDoctor img = new ImgDoctor ();
        jplmagen.add(img);
        jplmagen.repaint(); }
    /*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")
    private void jbAgregarActionPerformed(java.awt.event.ActionEvent evt) {
        String dni = jtfDNI.getText();
        boolean aux = doctor.existe("doctor", "dni_doctor", dni);
        if (aux==true){
            JOptionPane.showMessageDialog(rootPane, "El registro ya existe");
        }else{
            String dni_doctor = jtfDNI.getText();
            String nombre = jtfNombre.getText();
            String apellido= jtfApellido.getText();
            String email = jtfEmail.getText();
            String matricula = jtfMatricula.getText();
            String domicilio = jtfDomicilio.getText();
            String ciudad = jtfCiudad.getText();
            String telefono_fijo = jtfFijo.getText();
            String telefono_celular = jtfCelular.getText();
            String datos[] = {dni_doctor,nombre,apellido,email,matricula,domicilio,ciudad,telefono_fijo,telefono_celular};
            //array de datos , instruccion sql
            doctor.InsertarRegistro(datos, "insert into doctor values(?,?,?,?,?,?,?,?)"); }
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) { this.dispose(); }
    MySQL doctor = new MySQL();
    Object[][] dtDoctor;
    int fila = -1;
    // Variables declaration - do not modify
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JButton jbAgregar;
    private javax.swing.JButton jbCancelar;
    private javax.swing.JPanel jplmagen;
    private javax.swing.JPanel jp_duenio;
    private javax.swing.JTextField jtfApellido;
    private javax.swing.JTextField jtfCelular;
    private javax.swing.JTextField jtfCiudad;
    private javax.swing.JTextField jtfDNI;
    private javax.swing.JTextField jtfDomicilio;
    private javax.swing.JTextField jtfEmail;
    private javax.swing.JTextField jtfFijo;
    private javax.swing.JTextField jtfMatricula;
    private javax.swing.JTextField jtfNombre; // End of variables declaration
}
```


Paquete Reportes

BuscarHemograma.Java

```

package Reportes;
import Examenes.*;
import Examenes.*;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import ModuloUsuario.ImgDoctor;
import javax.swing.table.DefaultTableModel;
public class BuscarHemograma extends javax.swing.JInternalFrame {
private IniciarReporte jasper;
public BuscarHemograma() {
    initComponents();
    ImgDoctor img = new ImgDoctor();
    jplmagen.add(img);
    jplmagen.repaint();
    cargarTabla();
    jlBuscarCasoClinico.setVisible(false);
    jasper = new IniciarReporte();
}
private void cargarTabla(){
    String[] columnNames = {"Protocolo Lab","Caso Clinico","DNI del Duenio","Apellido","Nombre","Nombre Del Animal"};
    String colName[] = {"protocolo_lab","caso_clinico","dni_duenio","apellido","nombre","animal_pequenos.nombre"};
    //nombre de columnas, tabla, instruccion sql
    dtCasoClinico = casoClinico.GetTabla(colName, "consulta,duenio,animal_pequenos,doctor,hemograma", "SELECT * FROM
`consulta`,`duenio`,`animal_pequenos`,`doctor`,`hemograma` WHERE (`consulta`.`cod_animal`=`animal_pequenos`.`cod_animal`) AND
(`consulta`.`dni_doctor`=`doctor`.`dni_doctor`) AND (`animal_pequenos`.`dni_duenio`=`duenio`.`dni_duenio`)AND (`consulta`.`caso_clinico` =
`hemograma`.`caso_clinico`)");
    // se colocan los datos en la tabla
    DefaultTableModel datos = new DefaultTableModel(dtCasoClinico, columnNames);
    jtCasoClinico.setModel(datos);
    // se colocan los datos en la tabla
    int a = jtCasoClinico.getRowCount();
    //depurador de jtDuenio de item null
    int i=0;
    a--;
    while(a>=i){
        if(jtCasoClinico.getValueAt(i,1)==null){
            ((DefaultTableModel)jtCasoClinico.getModel()).removeRow(i);
            a--;
            i--;
        }
        i++;
    }
}
/*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")
private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
    int filamodificar= jtCasoClinico.getSelectedRow();
    if(filamodificar>=0){
        String protocolo_lab= jtCasoClinico.getValueAt(filamodificar, 0).toString();
        String reporte="Hemograma";
        jasper.ejecutarReporte(protocolo_lab,reporte);
    }
}
private void jtCasoClinicoMouseClicked(java.awt.event.MouseEvent evt) {
    fila=jtCasoClinico.rowAtPoint(evt.getPoint());
}
private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}
private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {

```

```

String texto= jTextField1.getText();
String[] columnNames = {"Protocolo Lab","Caso Clinico","DNI del Duenio","Apellido","Nombre","Nombre Del Animal"};
String colName[] = {"protocolo_lab","caso_clinico","dni_duenio","apellido","nombre","animal_pequenos.nombre"}; //nombre de columnas,
tabla, instruccion sql
if(jrbProtocoloLab.isSelected()==true){
    dtCasoClinico = casoClinico.GetTabla(colName, "consulta,duenio,animal_pequenos,doctor,hemograma", "SELECT * FROM
`consulta`,`duenio`,`animal_pequenos`,`doctor`,`hemograma` WHERE (`consulta`.`cod_animal`=`animal_pequenos`.`cod_animal`) AND
(`consulta`.`dni_doctor`=`doctor`.`dni_doctor`) AND (`animal_pequenos`.`dni_duenio`=`duenio`.`dni_duenio`)AND (`consulta`.`caso_clinico` =
`hemograma`.`caso_clinico`)AND(`hemograma`.`protocolo_lab`= \"\" + texto + \"\")");
}
if(jrbDni.isSelected()==true){
    dtCasoClinico = casoClinico.GetTabla(colName, "consulta,duenio,animal_pequenos,doctor,hemograma", "SELECT * FROM
`consulta`,`duenio`,`animal_pequenos`,`doctor`,`hemograma` WHERE (`consulta`.`cod_animal`=`animal_pequenos`.`cod_animal`) AND
(`consulta`.`dni_doctor`=`doctor`.`dni_doctor`) AND (`animal_pequenos`.`dni_duenio`=`duenio`.`dni_duenio`)AND (`consulta`.`caso_clinico` =
`hemograma`.`caso_clinico`)AND(`duenio`.`dni_duenio`= \"\" + texto + \"\")");
}
if(jrbNombreAnimal.isSelected()==true){
    dtCasoClinico = casoClinico.GetTabla(colName, "consulta,duenio,animal_pequenos,doctor,hemograma", "SELECT * FROM
`consulta`,`duenio`,`animal_pequenos`,`doctor`,`hemograma` WHERE (`consulta`.`cod_animal`=`animal_pequenos`.`cod_animal`) AND
(`consulta`.`dni_doctor`=`doctor`.`dni_doctor`) AND (`animal_pequenos`.`dni_duenio`=`duenio`.`dni_duenio`)AND (`consulta`.`caso_clinico` =
`hemograma`.`caso_clinico`)AND(`animal_pequenos`.`nombre`= \"\" + texto + \"\")");
}
DefaultTableModel datos = new DefaultTableModel(dtCasoClinico,columnNames);
jtCasoClinico.setModel(datos);// se colocan los datos en la tabla
//depurador de jtDuenio de item null
int a = jtCasoClinico.getRowCount();
int i=0;
a--;
while(a>=i){
    if(jtCasoClinico.getValueAt(i,1)==null){
        ((DefaultTableModel)jtCasoClinico.getModel()).removeRow(i);
        a--;
        i--;
    }
    i++;
}
}
Object[][] dtCasoClinico;
MySQL casoClinico = new MySQL();
int fila = -1;
// Variables declaration - do not modify
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JButton jbBuscar;
private javax.swing.JButton jbCancelar;
private javax.swing.JButton jbElegir;
public static javax.swing.JLabel jlBuscarCasoClinico;
private javax.swing.JPanel jPanelImagen;
private javax.swing.JRadioButton jrbDni;
private javax.swing.JRadioButton jrbNombreAnimal;
private javax.swing.JRadioButton jrbProtocoloLab;
public static javax.swing.JTable jtCasoClinico;
private javax.swing.JTextField jTextField1;
// End of variables declaration
}

```

BuscarQuimicaClinica.Java

```
package Reportes;
import Examenes.*;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import ModuloUsuario.ImgDoctor;
import javax.swing.table.DefaultTableModel;
public class BuscarQuimicaClinica extends javax.swing.JInternalFrame {
private IniciarReporte jasper;
    public BuscarQuimicaClinica() {
        initComponents();
        ImgDoctor img = new ImgDoctor();
        jplImagen.add(img);
        jplImagen.repaint();
        cargarTabla();
        jlBuscarCasoClinico.setVisible(false);
        jasper = new IniciarReporte();
    }
    private void cargarTabla(){
        String[] columnNames = {"Protocolo Lab", "Caso Clinico", "DNI del Duenio", "Apellido", "Nombre", "Nombre Del Animal"};
        String colName[] = {"protocolo_lab", "caso_clinico", "dni_duenio", "apellido", "nombre", "animal_pequenios.nombre"};
        //nombre de columnas, tabla, instruccion sql
        dtCasoClinico = casoClinico.GetTabla(colName, "consulta,duenio,animal_pequenios,doctor,quimica_clinica", "SELECT * FROM
`consulta`,`duenio`,`animal_pequenios`,`doctor`,`quimica_clinica` WHERE (`consulta`.`cod_animal`=`animal_pequenios`.`cod_animal`) AND
(`consulta`.`dni_doctor`=`doctor`.`dni_doctor`) AND (`animal_pequenios`.`dni_duenio`=`duenio`.`dni_duenio`)AND (`consulta`.`caso_clinico` =
`quimica_clinica`.`caso_clinico`)");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtCasoClinico, columnNames);
        jtCasoClinico.setModel(datos);
        // se colocan los datos en la tabla
        int a = jtCasoClinico.getRowCount();
        //depurador de jtDuenio de item null
        int i=0;
        a--;
        while(a>=i){
            if(jtCasoClinico.getValueAt(i,1)==null){
                ((DefaultTableModel)jtCasoClinico.getModel()).removeRow(i);
                a--;
                i--;
            }
            i++;
        }
    }
    /*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")
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar= jtCasoClinico.getSelectedRow();
        if(filamodificar>=0){
            String protocolo_lab= jtCasoClinico.getValueAt(filamodificar, 0).toString();
            String reporte="QuimicaClinica";
            jasper.ejecutarReporte(protocolo_lab,reporte);
        }
    }
    private void jtCasoClinicoMouseClicked(java.awt.event.MouseEvent evt) {
        fila=jtCasoClinico.rowAtPoint(evt.getPoint());
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String texto= jtfBuscar.getText();
        String[] columnNames = {"Protocolo Lab", "Caso Clinico", "DNI del Duenio", "Apellido", "Nombre", "Nombre Del Animal"};
        String colName[] = {"protocolo_lab", "caso_clinico", "dni_duenio", "apellido", "nombre", "animal_pequenios.nombre"};    //nombre de columnas,
        tabla, instruccion sql
        if(jrbProtocoloLab.isSelected()==true){
```

```

        dtCasoClinico = casoClinico.GetTabla(colName, "consulta,duenio,animal_pequenos,doctor,quimica_clinica", "SELECT * FROM
`consulta`,`duenio`,`animal_pequenos`,`doctor`,`quimica_clinica` WHERE (`consulta`.`cod_animal`=`animal_pequenos`.`cod_animal`) AND
(`consulta`.`dni_doctor`=`doctor`.`dni_doctor`) AND (`animal_pequenos`.`dni_duenio`=`duenio`.`dni_duenio`)AND (`consulta`.`caso_clinico` =
`quimica_clinica`.`caso_clinico`)AND(`quimica_clinica`.`protocolo_lab`='\" + texto + "\"");
    }
    if(jrbDni.isSelected()==true){
        dtCasoClinico = casoClinico.GetTabla(colName, "consulta,duenio,animal_pequenos,doctor,quimica_clinica", "SELECT * FROM
`consulta`,`duenio`,`animal_pequenos`,`doctor`,`quimica_clinica` WHERE (`consulta`.`cod_animal`=`animal_pequenos`.`cod_animal`) AND
(`consulta`.`dni_doctor`=`doctor`.`dni_doctor`) AND (`animal_pequenos`.`dni_duenio`=`duenio`.`dni_duenio`)AND (`consulta`.`caso_clinico` =
`quimica_clinica`.`caso_clinico`)AND(`duenio`.`dni_duenio`='\" + texto + "\"");
    }
    if(jrbNombreAnimal.isSelected()==true){
        dtCasoClinico = casoClinico.GetTabla(colName, "consulta,duenio,animal_pequenos,doctor,quimica_clinica", "SELECT * FROM
`consulta`,`duenio`,`animal_pequenos`,`doctor`,`quimica_clinica` WHERE (`consulta`.`cod_animal`=`animal_pequenos`.`cod_animal`) AND
(`consulta`.`dni_doctor`=`doctor`.`dni_doctor`) AND (`animal_pequenos`.`dni_duenio`=`duenio`.`dni_duenio`)AND (`consulta`.`caso_clinico` =
`quimica_clinica`.`caso_clinico`)AND(`animal_pequenos`.`nombre`='\" + texto + "\"");
    }
    DefaultTableModel datos = new DefaultTableModel(dtCasoClinico,columnNames);
    jtCasoClinico.setModel(datos);// se colocan los datos en la tabla
    //depurador de jtDuenio de item null
    int a = jtCasoClinico.getRowCount();
    int i=0;
    a--;
    while(a>=i){
        if(jtCasoClinico.getValueAt(i,1)==null){
            ((DefaultTableModel)jtCasoClinico.getModel()).removeRow(i);
            a--;
            i--;
        }
        i++;
    }
}
Object[][] dtCasoClinico;
MySQL casoClinico = new MySQL();
int fila = -1;
// Variables declaration - do not modify
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JButton jbBuscar;
private javax.swing.JButton jbCancelar;
private javax.swing.JButton jbElegir;
public static javax.swing.JLabel jlBuscarCasoClinico;
private javax.swing.JPanel jPanelImagen;
private javax.swing.JRadioButton jrbDni;
private javax.swing.JRadioButton jrbNombreAnimal;
private javax.swing.JRadioButton jrbProtocoloLab;
public static javax.swing.JTable jtCasoClinico;
private javax.swing.JTextField jtfBuscar;
// End of variables declaration
}

```

BuscarUrianalis.Java

```

package Reportes;
import Examenes.*;
import Clases.MySQL;
import Clases.PantallaPrincipal;
import ModuloUsuario.ImgDoctor;
import javax.swing.table.DefaultTableModel;
public class BuscarUrianalis extends javax.swing.JFrame {
private IniciarReporte jasper;
    public BuscarUrianalis() {
        initComponents();
        ImgDoctor img = new ImgDoctor();
        jplImagen.add(img);
        jplImagen.repaint();
        cargarTabla();
        jlBuscarCasoClinico.setVisible(false);
        jasper = new IniciarReporte();
    }
    private void cargarTabla(){
        String[] columnNames = {"Protocolo Lab", "Caso Clinico", "DNI del Duenio", "Apellido", "Nombre", "Nombre Del Animal"};
        String colName[] = {"protocolo_lab", "caso_clinico", "dni_duenio", "apellido", "nombre", "animal_pequenios.nombre"};
        //nombre de columnas, tabla, instruccion sql
        dtCasoClinico = casoClinico.GetTabla(colName, "consulta,duenio,animal_pequenios,doctor,urianalisis", "SELECT * FROM
`consulta`,`duenio`,`animal_pequenios`,`doctor`,`urianalisis` WHERE (`consulta`.`cod_animal` = `animal_pequenios`.`cod_animal`) AND
(`consulta`.`dni_doctor` = `doctor`.`dni_doctor`) AND (`animal_pequenios`.`dni_duenio` = `duenio`.`dni_duenio`)AND (`consulta`.`caso_clinico` =
`urianalisis`.`caso_clinico`)");
        // se colocan los datos en la tabla
        DefaultTableModel datos = new DefaultTableModel(dtCasoClinico, columnNames);
        jtCasoClinico.setModel(datos);
        // se colocan los datos en la tabla
        int a = jtCasoClinico.getRowCount();
        //depurador de jtDuenio de item null
        int i=0;
        a--;
        while(a>=i){
            if(jtCasoClinico.getValueAt(i,1)==null){
                ((DefaultTableModel)jtCasoClinico.getModel()).removeRow(i);
                a--;
                i--;
            }
            i++;
        }
    }
    /*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")
    private void jbElegirActionPerformed(java.awt.event.ActionEvent evt) {
        int filamodificar= jtCasoClinico.getSelectedRow();
        if(filamodificar>=0){
            String protocolo_lab= jtCasoClinico.getValueAt(filamodificar, 0).toString();
            String reporte="Urianalisis";
            jasper.ejecutarReporte(protocolo_lab,reporte);
        }
    }
    private void jtCasoClinicoMouseClicked(java.awt.event.MouseEvent evt) {
        fila=jtCasoClinico.rowAtPoint(evt.getPoint());
    }
    private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        this.dispose();
    }
    private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt) {
        String texto= jtfBuscar.getText();
        String[] columnNames = {"Protocolo Lab", "Caso Clinico", "DNI del Duenio", "Apellido", "Nombre", "Nombre Del Animal"};
        String colName[] = {"protocolo_lab", "caso_clinico", "dni_duenio", "apellido", "nombre", "animal_pequenios.nombre"};    //nombre de columnas,
        tabla, instruccion sql
        if(jrbProtocoloLab.isSelected()==true){

```

```

        dtCasoClinico = casoClinico.GetTabla(colName, "consulta,duenio,animal_pequenios,doctor,urianalisis", "SELECT * FROM
`consulta`,`duenio`,`animal_pequenios`,`doctor`,`urianalisis` WHERE (`consulta`.`cod_animal`=`animal_pequenios`.`cod_animal`) AND
(`consulta`.`dni_doctor`=`doctor`.`dni_doctor`) AND (`animal_pequenios`.`dni_duenio`=`duenio`.`dni_duenio`)AND (`consulta`.`caso_clinico` =
`urianalisis`.`caso_clinico`)AND(`urianalisis`.`protocolo_lab`= \"\" + texto + \"\")");
    }
    if(jrbDni.isSelected()==true){
        dtCasoClinico = casoClinico.GetTabla(colName, "consulta,duenio,animal_pequenios,doctor,urianalisis", "SELECT * FROM
`consulta`,`duenio`,`animal_pequenios`,`doctor`,`urianalisis` WHERE (`consulta`.`cod_animal`=`animal_pequenios`.`cod_animal`) AND
(`consulta`.`dni_doctor`=`doctor`.`dni_doctor`) AND (`animal_pequenios`.`dni_duenio`=`duenio`.`dni_duenio`)AND (`consulta`.`caso_clinico` =
`urianalisis`.`caso_clinico`)AND(`duenio`.`dni_duenio`= \"\" + texto + \"\")");
    }
    if(jrbNombreAnimal.isSelected()==true){
        dtCasoClinico = casoClinico.GetTabla(colName, "consulta,duenio,animal_pequenios,doctor,urianalisis", "SELECT * FROM
`consulta`,`duenio`,`animal_pequenios`,`doctor`,`urianalisis` WHERE (`consulta`.`cod_animal`=`animal_pequenios`.`cod_animal`) AND
(`consulta`.`dni_doctor`=`doctor`.`dni_doctor`) AND (`animal_pequenios`.`dni_duenio`=`duenio`.`dni_duenio`)AND (`consulta`.`caso_clinico` =
`urianalisis`.`caso_clinico`)AND(`animal_pequenios`.`nombre`= \"\" + texto + \"\")");
    }
    DefaultTableModel datos = new DefaultTableModel(dtCasoClinico,columnNames);
    jtCasoClinico.setModel(datos);// se colocan los datos en la tabla
    //depurador de jtDuenio de item null
    int a = jtCasoClinico.getRowCount();
    int i=0;
    a--;
    while(a>=i){
        if(jtCasoClinico.getValueAt(i,1)==null){
            ((DefaultTableModel)jtCasoClinico.getModel()).removeRow(i);
            a--;
            i--;
        }
        i++;
    }
}
Object[][] dtCasoClinico;
MySQL casoClinico = new MySQL();
int fila = -1;
// Variables declaration - do not modify
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JButton jbBuscar;
private javax.swing.JButton jbCancelar;
private javax.swing.JButton jbElegir;
public static javax.swing.JLabel jlBuscarCasoClinico;
private javax.swing.JPanel jPanelImagen;
private javax.swing.JRadioButton jrbDni;
private javax.swing.JRadioButton jrbNombreAnimal;
private javax.swing.JRadioButton jrbProtocoloLab;
public static javax.swing.JTable jtCasoClinico;
private javax.swing.JTextField jtfBuscar;
// End of variables declaration
}

```

ListadoMostrarRetornoAP.Java

```

package Reportes;
import Clases.MySQL;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import javax.swing.DefaultComboBoxModel;
public class EstadisticaAnálisisDocenteAlumno extends javax.swing.JFrame {
private IniciarReporte2 jasper;
private IniciarReporte3 jasper2;
    public EstadisticaAnálisisDocenteAlumno() {
        initComponents();
        jcbLista.setVisible(false);
        jlApellidoNombre.setVisible(false);
        jDateChooser1.setVisible(false);
        jDateChooser2.setVisible(false);
        jDateChooser1.setDate(new Date());
        jDateChooser2.setDate(new Date());
        String dia = Integer.toString(jDateChooser1.getCalendar().get(Calendar.DAY_OF_MONTH));
        String mes = Integer.toString(jDateChooser1.getCalendar().get(Calendar.MONTH) + 1);
        String year = Integer.toString(jDateChooser1.getCalendar().get(Calendar.YEAR));
        String fecha = (year + "-" + mes + "-" + dia);
        jlFecha1.setText("2015-1-1");
        jlFecha2.setText(fecha);
        jasper = new IniciarReporte2();
        jasper2 = new IniciarReporte3();
    }
    private void cargarLista(){
        dcLista = lista.GetColumna("doctor", "dni_doctor", "SELECT * FROM doctor");
        // se colocan los datos en la tabla
        DefaultComboBoxModel datos = new DefaultComboBoxModel(dcLista);
        jcbLista.setModel(datos);
        int a = jcbLista.getItemCount();
        //depurador de jcbRaza de item null
        int i=0;
        a--;
        while(a>=i){
            if(jcbLista.getItemAt(i)==null){
                jcbLista.removeItemAt(i);
                a--;
                i--;
            }
            i++;
        }
        doctorAP();
    }
    private void cargarLista2(){
        dcLista = lista.GetColumna("alumno", "dni_alumno", "SELECT * FROM alumno");
        // se colocan los datos en la tabla
        DefaultComboBoxModel datos = new DefaultComboBoxModel(dcLista);
        jcbLista.setModel(datos);
        int a = jcbLista.getItemCount();
        //depurador de jcbRaza de item null
        int i=0;
        a--;
        while(a>=i){
            if(jcbLista.getItemAt(i)==null){
                jcbLista.removeItemAt(i);
                a--;
                i--;
            }
            i++;
        }
        alumnoAP();
    }
}

```



```

private void doctorAP(){
    String dni= (String)jcbLista.getSelectedItemAt();
    String apellido = lista.GetData("apellido", "SELECT apellido FROM doctor WHERE dni_doctor = \"\"+dni+\"\"");
    String nombre = lista.GetData("nombre", "SELECT nombre FROM doctor WHERE dni_doctor = \"\"+dni+\"\"");
    jlApellidoNombre.setText(apellido+" "+ nombre);
}
private void alumnoAP(){
    String dni= (String)jcbLista.getSelectedItemAt();
    String apellido = lista.GetData("apellido", "SELECT apellido FROM alumno WHERE dni_alumno = \"\"+dni+\"\"");
    String nombre = lista.GetData("nombre", "SELECT nombre FROM alumno WHERE dni_alumno = \"\"+dni+\"\"");
    jlApellidoNombre.setText(apellido+" "+ nombre);
}
private void fechaLabel (){
    String comparar =jcbLista3.getSelectedItemAt().toString();
    String comparar2 ="Todas las Fechas";
    String comparar3 ="Un Año";
    String comparar4 ="Un Mes";
    String comparar5 ="Una Semana";
    String comparar6 ="Un Dia";
    String comparar7 ="Tiempo Especifico";
    String dia = Integer.toString(jDateChooser1.getCalendar().get(Calendar.DAY_OF_MONTH));
    String mes = Integer.toString(jDateChooser1.getCalendar().get(Calendar.MONTH) + 1);
    String year = Integer.toString(jDateChooser1.getCalendar().get(Calendar.YEAR));
    String fecha = (year + "-" + mes+ "-" + dia);
    DateFormat fecha0 = new SimpleDateFormat("yyyy-MM-dd");
    Date nacio = jDateChooser2.getDate();
    if(comparar2==comparar){
        jDateChooser1.setVisible(false);
        jDateChooser2.setVisible(false);
        jlFecha1.setText("1990-1-1");
        jlFecha2.setText("2050-1-1");
    }
    if(comparar3==comparar){
        jDateChooser1.setVisible(true);
        jDateChooser2.setVisible(false);
        jlFecha1.setText(year+"-1-1");
        jlFecha2.setText(year+"-12-31");
    }
    if(comparar4==comparar){
        jDateChooser1.setVisible(true);
        jDateChooser2.setVisible(false);
        jlFecha1.setText(year + "-" + mes+ "-1");
        dia = Integer.toString(jDateChooser1.getCalendar().getActualMaximum(Calendar.DAY_OF_MONTH));
        jlFecha2.setText(year + "-" + mes+ "-" + dia);
    }
    if(comparar5==comparar){
        jDateChooser1.setVisible(true);
        jDateChooser2.setVisible(false);
        int diaSemana = jDateChooser1.getCalendar().get(Calendar.DAY_OF_WEEK);
        int diaSemana2 = 8 - diaSemana;
        int diaSemana3 = diaSemana-2;
        if(diaSemana==1){
            diaSemana2=0;
            diaSemana3=6;
        }
        nacio = jDateChooser1.getDate();
        Calendar fecha3 = Calendar.getInstance();
        fecha3.setTime(nacio);
        fecha3.add(Calendar.DATE, -diaSemana3 );
        SimpleDateFormat format1 = new SimpleDateFormat("yyyy-MM-dd");
        String formatted = format1.format(fecha3.getTime());
        jlFecha1.setText(formatted.toString());
        fecha3.setTime(nacio);
        fecha3.add(Calendar.DATE, diaSemana2 );
        formatted = format1.format(fecha3.getTime());
        jlFecha2.setText(formatted.toString());
    }
    if(comparar6==comparar){
        jDateChooser1.setVisible(true);
        jDateChooser2.setVisible(false);
        jlFecha1.setText(fecha);
    }
}

```

```
        jlFecha2.setText(fecha);
    }
    if(comparar7==comparar){
        jDateChooser1.setVisible(true);
        jDateChooser2.setVisible(true);
        jlFecha1.setText(fecha);
        jlFecha2.setText(fecha0.format(nacio));
    }
}

/*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")
private void jcbLista2ItemStateChanged(java.awt.event.ItemEvent evt) {
    jcbLista.setVisible(false);
    jlApellidoNombre.setVisible(false);
    String comparar =jcbLista2.getSelectedItem().toString();
    String comparar2 ="Solo un Docente";
    String comparar3 ="Solo un Alumno";
    if(comparar2==comparar){
        cargarLista();
        jcbLista.setVisible(true);
        jlApellidoNombre.setVisible(true);
    }
    if(comparar3==comparar){
        cargarLista2();
        jcbLista.setVisible(true);
        jlApellidoNombre.setVisible(true);
    }
}

private void jcbListaItemStateChanged(java.awt.event.ItemEvent evt) {
    String comparar =jcbLista2.getSelectedItem().toString();
    String comparar2 ="Solo un Docente";
    String comparar3 ="Solo un Alumno";
    if(comparar2==comparar){
        doctorAP();
    }
    if(comparar3==comparar){
        alumnoAP();
    }
}

private void jDateChooser1PropertyChange(java.beans.PropertyChangeEvent evt) {
    fechaLabel();
}

private void jDateChooser2PropertyChange(java.beans.PropertyChangeEvent evt) {
    fechaLabel();
}

private void jcbLista3ItemStateChanged(java.awt.event.ItemEvent evt) {
    fechaLabel();
}

private void jbReporteActionPerformed(java.awt.event.ActionEvent evt) {
    String fecha1 = jlFecha1.getText();
    String fecha2 = jlFecha2.getText();
    String comparar =jcbLista2.getSelectedItem().toString();
    String comparar2 ="Reporte Todos Docentes y alumnos";
    String comparar3 ="Informe Todos Docentes y alumnos";
    String comparar4 ="Informe Todos Animales";
    String comparar5 ="Solo un Docente";
    String comparar6 ="Solo un Alumno";
    if(comparar2==comparar){
        String reporte="CasoClinicoTodosAlumnos";
        jasper2.ejecutarReporte(fecha1,fecha2,reporte);
    }
    if(comparar3==comparar){
        String reporte="CasoClinicoInformeTodos";
        jasper2.ejecutarReporte(fecha1,fecha2,reporte);
    }
    if(comparar4==comparar){
        String reporte="CasoClinicoInformeAnimales";
        jasper2.ejecutarReporte(fecha1,fecha2,reporte);
    }
}
```



```
if(comparar5==comparar){
String reporte="CasoClinico1Docente";
String dni= jcbLista.getSelectedItem().toString();
jasper.ejecutarReporte(dni,fecha1,fecha2,reporte);
}
if(comparar6==comparar){
String reporte="CasoClinico1Alumno";
String dni= jcbLista.getSelectedItem().toString();
jasper.ejecutarReporte(dni,fecha1,fecha2,reporte);
}
}
MySQL lista = new MySQL();
Object[] dcLista;
Object[] dcLista2;
// Variables declaration - do not modify
private com.toedter.calendar.JDateChooser jDateChooser1;
private com.toedter.calendar.JDateChooser jDateChooser2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel5;
private javax.swing.JButton jbReporte;
private javax.swing.JComboBox jcbLista;
private javax.swing.JComboBox jcbLista2;
private javax.swing.JComboBox jcbLista3;
private javax.swing.JLabel jlApellidoNombre;
private javax.swing.JLabel jlFecha1;
private javax.swing.JLabel jlFecha2;
// End of variables declaration
}
```

IniciarReporte.Java

```
package Reportes;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.HashMap;
import java.util.Map;
import net.sf.jasperreports.engine.*;
import net.sf.jasperreports.engine.util.JRLoader;
import net.sf.jasperreports.view.*;
import net.sf.jasperreports.view.save.JRPdfSaveContributor.*;
import net.sf.jasperreports.view.JRViewer;
import net.sf.jasperreports.view.save.JRMultipleSheetsXlsSaveContributor.*;
public class IniciarReporte {
    static String bd = "siga";
    static String login = "root";
    static String password = "";
    static String url = "jdbc:mysql://localhost/" + bd;
    private static String logo = System.getProperty("user.dir") + "\\src\\Imagenes\\escudoUCHA2.jpg";
    Connection conn = null;
    public IniciarReporte(){
        try{
            //obtenemos el driver de para mysql
            Class.forName("com.mysql.jdbc.Driver");
            //obtenemos la conexión
            conn = DriverManager.getConnection(url,login,password);
            if (conn!=null){
                System.out.println("Conección a base de datos "+bd+" listo");
            }
        }catch(SQLException e){
            System.out.println(e);
        }catch(ClassNotFoundException e){
            System.out.println(e);
        }
    }
    public void ejecutarReporte(String protocolo_lab,String reporte){
        try{
            //String archivo = "D:\\Users\\Usuario\\Documents\\NetBeansProjects\\Siga1\\src\\Reportes\\" + reporte + ".jasper";
            String archivo = System.getProperty("user.dir") + "\\src\\Reportes\\" + reporte + ".jasper";
            //InputStream archivo = getClass().getClassLoader().getResourceAsStream("Reportes/" + reporte + ".jasper");
            System.out.println("Cargando desde: " + archivo);
            if(archivo == null){
                System.out.println("No se encuentra el archivo. ");
                System.exit(2);
            }
            JasperReport masterReport = null;
            try{
                masterReport =(JasperReport)JRLoader.loadObject(archivo);
            }catch(JRException e){
                System.out.println("Error cargando el reporte maestro: " + e.getMessage());
                System.exit(3);
            }
            //este es el parametro
            Map parametro = new HashMap();
            parametro.put("protocolo_lab",protocolo_lab);

            parametro.put("logo",logo);
            //reporte diseñado y compilado con ireport
            JasperPrint jasperPrint = JasperFillManager.fillReport(masterReport,parametro, conn);
            //Se lanza el viewer de jasper, no termina aplicacion al salir
            JasperViewer jviewer = new JasperViewer(jasperPrint,false);
            jviewer.setTitle("Geniz-Report");
            jviewer.setVisible(true);
        }catch(Exception j){
            System.out.println("Mensaje de Error:" + j.getMessage());
        }
    }
}
```

```
    }  
}  
public void cerrar(){  
    try{  
        conn.close();  
    }catch(SQLException ex){  
        ex.printStackTrace();  
    }  
}
```

IniciarReporte2.Java

```
package Reportes;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.SQLDataException;
import javax.swing.*;
import java.util.HashMap;
import java.util.Map;
import net.sf.jasperreports.engine.*;
import net.sf.jasperreports.engine.util.JRLoader;
import net.sf.jasperreports.view.*;
import net.sf.jasperreports.view.save.JRPdfSaveContributor.*;
import net.sf.jasperreports.view.save.JRMultipleSheetsXlsSaveContributor.*;
public class IniciarReporte2 {
    static String bd = "siga";
    static String login = "root";
    static String password = "";
    static String url = "jdbc:mysql://localhost/" + bd;
    private static String logo = System.getProperty("user.dir") + "\\src\\Imagenes\\escudoUCHA2.jpg";
    Connection conn = null;
    public IniciarReporte2(){
        try{
            //obtenemos el driver de para mysql
            Class.forName("com.mysql.jdbc.Driver");
            //obtenemos la conexión
            conn = DriverManager.getConnection(url,login,password);
            if (conn!=null){
                System.out.println("Conección a base de datos "+bd+" listo");
            }
        }catch(SQLException e){
            System.out.println(e);
        }catch(ClassNotFoundException e){
            System.out.println(e);
        }
    }
    public void ejecutarReporte(String dni,String fecha1,String fecha2,String reporte){
        try{
            String archivo = System.getProperty("user.dir") + "\\src\\Reportes\\" + reporte + ".jasper";
            System.out.println("Cargando desde: " + archivo);
            if(archivo == null){
                System.out.println("No se encuentra el archivo. ");
                System.exit(2);
            }
            JasperReport masterReport = null;
            try{
                masterReport =(JasperReport)JRLoader.loadObject(archivo);
            }catch(JRException e){
                System.out.println("Error cargando el reporte maestro: "+ e.getMessage());
                System.exit(3);
            }
            //este es el parametro
            Map parametro = new HashMap();
            parametro.put("dni",dni);
            parametro.put("fecha1",fecha1);
            parametro.put("fecha2",fecha2);
            parametro.put("logo",logo);
            //reporte diseñado y compilado con ireport
            JasperPrint jasperPrint = JasperFillManager.fillReport(masterReport,parametro, conn);
            //Se lanza el viewer de jasper, no termina aplicacion al salir
            JasperViewer jviewer = new JasperViewer(jasperPrint,false);
            jviewer.setTitle("Geniz-Report");
            jviewer.setVisible(true);
        }catch(Exception j){
            System.out.println("Mensaje de Error: " + j.getMessage());
        }
    }
}
```

```
}  
public void cerrar(){  
    try{  
        conn.close();  
    }catch(SQLException ex){  
        ex.printStackTrace();  
    }  
}  
}
```


IniciarReporte3.Java

```

package Reportes;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.SQLDataException;
import javax.swing.*;
import java.util.HashMap;
import java.util.Map;
import net.sf.jasperreports.engine.*;
import net.sf.jasperreports.engine.util.JRLoader;
import net.sf.jasperreports.view.*;
import net.sf.jasperreports.view.save.JRPdfSaveContributor.*;
import net.sf.jasperreports.view.JRViewer;
import net.sf.jasperreports.view.save.JRMultipleSheetsXlsSaveContributor.*;
public class IniciarReporte3 {
    static String bd = "siga";
    static String login = "root";
    static String password = "";
    static String url = "jdbc:mysql://localhost/" + bd;
    private static String logo = System.getProperty("user.dir") + "\\src\\Imágenes\\escudoUCHA2.jpg";
    Connection conn = null;
    public IniciarReporte3(){
        try{
//obtenemos el driver de para mysql
            Class.forName("com.mysql.jdbc.Driver");
//obtenemos la conexión
            conn = DriverManager.getConnection(url,login,password);
            if (conn!=null){
                System.out.println("Conección a base de datos "+bd+" listo");
            }
        }catch(SQLException e){
            System.out.println(e);
        }catch(ClassNotFoundException e){
            System.out.println(e);
        }
    }
    public void ejecutarReporte(String fecha1,String fecha2,String reporte){
        try{
            String archivo = System.getProperty("user.dir") + "\\src\\Reportes\\" + reporte + ".jasper";
            System.out.println("Cargando desde: " + archivo);
            if(archivo == null){
                System.out.println("No se encuentra el archivo. ");
                System.exit(2);
            }
            JasperReport masterReport = null;
            try{
                masterReport =(JasperReport)JRLoader.loadObject(archivo);
            }catch(JRException e){
                System.out.println("Error cargando el reporte maestro: " + e.getMessage());
                System.exit(3);
            }
            //este es el parametro
            Map parametro = new HashMap();
            parametro.put("fecha1",fecha1);
            parametro.put("fecha2",fecha2);
            parametro.put("logo",logo);
            //reporte diseñado y compilado con ireport
            JasperPrint jasperPrint = JasperFillManager.fillReport(masterReport,parametro, conn);
            //Se lanza el viewer de jasper, no termina aplicacion al salir
            JasperViewer jviewer = new JasperViewer(jasperPrint,false);
            jviewer.setTitle("Geniz-Report");
            jviewer.setVisible(true);
        }catch(Exception j){
            System.out.println("Mensaje de Error: " + j.getMessage());
        }
    }
}

```



```
public void cerrar(){  
    try{  
        conn.close();  
    }catch(SQLException ex){  
        ex.printStackTrace();  
    }  
}
```

ReporteHemograma.Java

```
package Reportes;
public class ReporteHemograma extends javax.swing.JInternalFrame {
    private IniciarReporte jasper;
    public ReporteHemograma() {
        initComponents();
        jasper = new IniciarReporte();
    }
    /*This method is called from within the constructor to initialize the form.
    * WARNING: Do NOT modify this code. The content of this method is always
    * regenerated by the Form Editor.*/
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jtfProtocoloLab = new javax.swing.JTextField();
        jbRealizar = new javax.swing.JButton();

        jLabel1.setText("Protocolo Laboratorio:");

        jbRealizar.setText("Realizar Reporte");
        jbRealizar.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jbRealizarActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(37, 37, 37)
                    .addComponent(jLabel1)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                    .addComponent(jtfProtocoloLab, javax.swing.GroupLayout.PREFERRED_SIZE, 106, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                    .addComponent(jbRealizar)
                    .addContainerGap(15, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(63, 63, 63)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(jLabel1)
                        .addComponent(jtfProtocoloLab, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jbRealizar))
                    .addContainerGap(192, Short.MAX_VALUE))
        );
        pack();
    } // </editor-fold>
    private void jbRealizarActionPerformed(java.awt.event.ActionEvent evt) {
        String protocolo_lab = jtfProtocoloLab.getText();
        String reporte="Hemograma";
        jasper.ejecutarReporte(protocolo_lab,reporte);
    }
    // Variables declaration - do not modify
    private javax.swing.JLabel jLabel1;
    private javax.swing.JButton jbRealizar;
    private javax.swing.JTextField jtfProtocoloLab; // End of variables declaration }
}
```