

ANEXO C

CÓDIGOS FUENTE DE LA APLICACIÓN DESARROLLADA PARA EL HUVR

C.1. Esqueleto de la base de datos

Aquí vamos a ver los comandos SQL necesarios para construir el esqueleto de la base de datos del Proyecto. Dichos comandos se encuentran recogidos en el fichero “esqueleto.sql”.

Fichero esqueleto.sql

```
CREATE TABLE paciente (id VARCHAR(10),
    nombre VARCHAR(64),
    apellidos VARCHAR(256),
    domicilio VARCHAR(256),
    localidad VARCHAR(40),
    telefono VARCHAR(16),
    fecnac DATE,
    numss CHAR(12),
    resenf TEXT,
    resexpl TEXT,
    expcomp TEXT,
    diagnost TEXT,
    tratamto TEXT,
    fundclin TEXT,
    diabet CHAR(1),
    hiperten CHAR(1),
    anticoag CHAR(1),
    alermed TEXT,
    enfants TEXT,
    otros TEXT,
    PRIMARY KEY (id));

CREATE TABLE prueba (puid VARCHAR(64),
    fecha DATE,
    hora TIME,
    medico VARCHAR(256),
    descripcion TEXT,
    id VARCHAR(10),
    PRIMARY KEY (puid), FOREIGN KEY (id) REFERENCES paciente ON
DELETE CASCADE);

CREATE TABLE serie (suid VARCHAR(64),
    tipo VARCHAR(16),
    fecha DATE,
    hora TIME,
    puid VARCHAR(64),
    PRIMARY KEY (suid), FOREIGN KEY (puid) REFERENCES prueba ON
DELETE CASCADE);

CREATE TABLE fichero (nombre VARCHAR(2048),
    ancho INTEGER,
    alto INTEGER,
    suid VARCHAR(64),
    PRIMARY KEY (nombre), FOREIGN KEY (suid) REFERENCES serie ON
DELETE CASCADE);
```

```
CREATE TABLE fecha_operacion (medico VARCHAR(256),
    fecha DATE,
    hora TIME,
    texto DATE,
    puid VARCHAR(64),
    PRIMARY KEY (medico, fecha, hora),FOREIGN KEY (puid) REFERENCES
prueba ON DELETE CASCADE);

CREATE TABLE biopsia (medico VARCHAR(256),
    fecha DATE,
    hora TIME,
    texto TEXT,
    puid VARCHAR(64),
    PRIMARY KEY (medico, fecha, hora),FOREIGN KEY (puid) REFERENCES
prueba ON DELETE CASCADE);

CREATE TABLE recomendaciones_al_alta ( medico VARCHAR(256),
    fecha DATE,
    hora TIME,
    texto TEXT,
    puid VARCHAR(64),
    PRIMARY KEY ( medico, fecha, hora),FOREIGN KEY (puid) REFERENCES
prueba ON DELETE CASCADE);
```

C.2. Aplicación de contestación a la Gestoría de Usuarios

Aquí veremos todas las clases JAVA que se implementaron para esta aplicación.

Clase HUVR.java

```
/**
 * Esta clase es la principal y ejecutable; contiene los métodos a
 * llamar desde el exterior
 * @autora Ma José Aguilar Porro
 */

package huvr;

import huvr.menus.MenuPpal;
import huvr.paneles.PanelPpal;
import huvr.nucleo.FuncionesGUI;
import huvr.escritores.escriptorError;
import javax.swing.*.*;

public class HUVR extends javax.swing.JFrame{

    /** Cuerpo de la aplicación gráfica */
    public PanelPpal PanelPrincipal = null;
    /** menú de la aplicación gráfica */
    public MenuPpal MenuPrincipal = null;

    /**
     * MAIN PRINCIPAL.
     */

    public static void main (String args[]){
        try{
```

```
HUVR _huvr = new HUVR(args); // creamos una instancia
if (_huvr.MenuPrincipal == null) {
    // si no se lanzó la GUI
    new FuncionesGUI().informar("Saliendo...\n");
    System.exit(0);
}
}
catch (Throwable e){
    new FuncionesGUI().informar(
        "main(): " + new escritorError().escribirError(e));
}
}

/**
 * Constructor sin parámetros. Ajusta su referencia en
 * FuncionesGUI.
 */

private HUVR(){
    try{
        new FuncionesGUI().fijarFrameRaiz(this);
        // así la encontraremos si lo necesitamos
    }
    catch (Throwable e){
        new FuncionesGUI().informar("HUVR(): " + new
            escritorError().escribirError(e));
    }
}

/**
 * Constructor con argumentos, se ocupara de lanzar la interfaz
 * de usuario
 */

public HUVR(String[] args){
    this();
    try{
        if (args != null && args.length != 0)
            System.out.println("saliendo...");
        else
            this.interfazGrafica(); // lanzamos la interfaz de usuario
    }
    catch (Throwable e){
        new FuncionesGUI().informar("CMA(String[]): " + new
            escritorError().escribirError(e));
    }
}

/**
 * Este método es el que inicializa toda la interfaz gráfica
 */

private void interfazGrafica(){
    try{
        //título de la ventana
        setTitle("Formulario de contestación a la Gestoría de
            Usuarios");
        // gestión del cierre de la ventana por la "x"
        addWindowListener(new java.awt.event.WindowAdapter(){
            public void WindowClosing(java.awt.event.WindowEvent evt){
                System.exit(0);}});
    }
}
```

```
// fijamos la apariencia como la del sistema donde se ejecuta
try{
    UIManager.setLookAndFeel(javax.swing.UIManager.
        getSystemLookAndFeelClassName());
    SwingUtilities.updateComponentTreeUI(this);
}
catch (Throwable e){
    new FuncionesGUI().informar(new escritorError().
        escribirError(e));
}

// la barra de menús
MenuPrincipal = new MenuPpal();
setJMenuBar(MenuPrincipal);
// el contenido
PanelPrincipal = new PanelPpal();
getContentPane().add(PanelPrincipal);

//ajustamos sitio y tamaño de los componentes
pack();
setResizable(false);
//mostramos
show();
}
catch (Throwable e){
    System.exit(0);
}
}
```

Clase paneles.PanelPpal.java

```
/**
 * Esta clase define el cuerpo de la aplicación gráfica
 */

package huvr.paneles;

import javax.swing.*;

public class PanelPpal extends JPanel{

    // Ponemos los subpaneles como variables públicas para su acceso
    // externo

    /** panel de datos del paciente */
    public PanelDatosPersonales _PanelDatosPersonales;
    /** panel de datos médicos */
    public PanelDatosClinicos _PanelDatosClinicos;

    /**
     * Constructor
     */

    public PanelPpal(){

        /** Definimos cada uno de los paneles */
        _PanelDatosPersonales = new PanelDatosPersonales();
        _PanelDatosClinicos = new PanelDatosClinicos();

        // ubicamos cada componente en su sitio
```

```
setLayout(new BorderLayout(this, BorderLayout.Y_AXIS));

JPanel subpanel1 = new JPanel();
subpanel1.add(_PanelDatosPersonales);
subpanel1.setBorder(BorderFactory.createTitledBorder(
    "Datos Personales del Paciente"));
add(subpanel1);

JPanel subpanel2 = new JPanel();
subpanel2.add(_PanelDatosClinicos);
subpanel2.setBorder(BorderFactory.createTitledBorder(
    "Datos clínicos"));
add(subpanel2);
}
}
```

Clase paneles.PanelDatosPersonales.java

```
/**
 * Panel de datos personales del paciente
 */

package huvr.paneles;

import javax.swing.*;
import java.awt.*;

public class PanelDatosPersonales extends JPanel{
    // Ponemos los componentes como variables públicas para su acceso
    // exterior. Para los campos de texto, voy agrupando campo + etiqueta

    public static JTextField nombreCampo;
    public JLabel nombre;
    public static JTextField apell1Campo;
    public JLabel apell1;
    public static JTextField apell2Campo;
    public JLabel apell2;
    public static JTextField domicilioCampo;
    public JLabel domicilio;
    public static JTextField localidadCampo;
    public JLabel localidad;
    public static JTextField tfnoCampo;
    public JLabel numTfno;
    public static JTextField fecNac1Campo;
    public static JTextField fecNac2Campo;
    public static JTextField fecNac3Campo;
    public JLabel fecNac;
    public static JTextField afiliSSCampo;
    public JLabel afiliSSocial;
    public static JTextField taSSCampo;
    public JLabel taSSocial;
    public static CheckboxGroup apto;
    public static Checkbox si;
    public static Checkbox no;

    /**
     * Constructor de la clase
     */

    public PanelDatosPersonales(){
```

```
nombreCampo = new JTextField("");
nombreCampo.setEnabled(true);
nombreCampo.setToolTipText("Inserte el nombre del paciente");
nombre = new JLabel("Nombre");
nombre.setToolTipText("Inserte el nombre del paciente");
nombre.setLabelFor(nombreCampo);
apell1Campo = new JTextField("");
apell1Campo.setEnabled(true);
apell1Campo.setToolTipText("Inserte el primer apellido");
apell1 = new JLabel("Primer Apellido");
apell1.setToolTipText("Inserte el primer apellido");
apell1.setLabelFor(apell1Campo);
apell2Campo = new JTextField("");
apell2Campo.setEnabled(true);
apell2Campo.setToolTipText("Inserte el segundo apellido");
apell2 = new JLabel("Segundo Apellido");
apell2.setToolTipText("Inserte el segundo apellido");
apell2.setLabelFor(apell2Campo);
domicilioCampo = new JTextField("");
domicilioCampo.setEnabled(true);
domicilioCampo.setToolTipText("Inserte el domicilio");
domicilio = new JLabel("Domicilio");
domicilio.setToolTipText("Inserte el domicilio");
domicilio.setLabelFor(domicilioCampo);
localidadCampo = new JTextField("");
localidadCampo.setEnabled(true);
localidadCampo.setToolTipText("Inserte la localidad");
localidad = new JLabel("Localidad");
localidad.setToolTipText("Inserte la localidad");
localidad.setLabelFor(localidadCampo);
tfnoCampo = new JTextField("");
tfnoCampo.setEnabled(true);
tfnoCampo.setToolTipText("Inserte el teléfono");
numTfno = new JLabel("Teléfono");
numTfno.setToolTipText("Inserte el teléfono");
numTfno.setLabelFor(tfnoCampo);
fecNac1Campo = new JTextField(2);
fecNac1Campo.setEnabled(true);
fecNac1Campo.setToolTipText("Insértela de la siguiente forma:
    p.ej: 7 de mayo de 1977: 07/05/1977");
fecNac2Campo = new JTextField(2);
fecNac2Campo.setEnabled(true);
fecNac2Campo.setToolTipText("Insértela de la siguiente forma:
    p.ej: 7 de mayo de 1977: 07/05/1977");
fecNac3Campo = new JTextField(4);
fecNac3Campo.setEnabled(true);
fecNac3Campo.setToolTipText("Insértela de la siguiente forma:
    p.ej: 7 de mayo de 1977: 07/05/1977");
fecNac = new JLabel("Fecha de Nacimiento");
fecNac.setToolTipText("Insértela de la siguiente forma: p.ej: 7 de
    mayo de 1977: 07051977");
fecNac.setLabelFor(fecNac1Campo);
afiliSSCampo = new JTextField("");
afiliSSCampo.setEnabled(true);
afiliSSCampo.setToolTipText("Inserte el número sin barras ni
    espacios");
afiliSSSocial = new JLabel("N° de la S. Social");
afiliSSSocial.setToolTipText("Inserte el número sin barras ni
    espacios");
afiliSSSocial.setLabelFor(afiliSSCampo);
```

```
taSSCampo = new JTextField("");
taSSCampo.setEnabled(true);
taSSCampo.setToolTipText("Nº sin barras ni espacios, si el
    paciente no tiene, inserte el nº de la SS");
taSSocial = new JLabel("Nº de Tarjeta Sanitaria");
taSSocial.setToolTipText("Nº sin barras ni espacios, si el
    paciente no tiene, inserte el nº de la SS");
taSSocial.setLabelFor(taSSCampo);
apto = new CheckboxGroup();
si = new Checkbox("Sí", false, apto);
no = new Checkbox("No", false, apto);

// A continuación, colocamos los componentes en la interfaz

JPanel panel_intermedio = new JPanel();
panel_intermedio.setLayout(new BoxLayout(panel_intermedio,
    BoxLayout.Y_AXIS));

JPanel subpanel_1 = new JPanel();
subpanel_1.setLayout(new BoxLayout(subpanel_1, BoxLayout.X_AXIS));
subpanel_1.add(nombre);
subpanel_1.add(Box.createRigidArea(new Dimension(5,0)));
subpanel_1.add(nombreCampo);
subpanel_1.add(Box.createRigidArea(new Dimension(10,0)));
subpanel_1.add(apell1);
subpanel_1.add(Box.createRigidArea(new Dimension(5,0)));
subpanel_1.add(apell1Campo);
subpanel_1.add(Box.createRigidArea(new Dimension(10,0)));
subpanel_1.add(apell2);
subpanel_1.add(Box.createRigidArea(new Dimension(5,0)));
subpanel_1.add(apell2Campo);
panel_intermedio.add(subpanel_1);

JPanel subpanel_2 = new JPanel();
subpanel_2.setLayout(new BoxLayout(subpanel_2, BoxLayout.X_AXIS));
subpanel_2.add(domicilio);
subpanel_2.add(Box.createRigidArea(new Dimension(5,0)));
subpanel_2.add(domicilioCampo);
subpanel_2.add(Box.createRigidArea(new Dimension(10,0)));
subpanel_2.add(localidad);
subpanel_2.add(Box.createRigidArea(new Dimension(5,0)));
subpanel_2.add(localidadCampo);
panel_intermedio.add(subpanel_2);

JPanel subpanel_3 = new JPanel();
subpanel_3.setLayout(new BoxLayout(subpanel_3, BoxLayout.X_AXIS));
subpanel_3.add(numTfno);
subpanel_3.add(Box.createRigidArea(new Dimension(5,0)));
subpanel_3.add(tfnoCampo);
subpanel_3.add(Box.createRigidArea(new Dimension(10,0)));
subpanel_3.add(afiliSSocial);
subpanel_3.add(Box.createRigidArea(new Dimension(5,0)));
subpanel_3.add(afiliSSCampo);
subpanel_3.add(Box.createRigidArea(new Dimension(10,0)));
subpanel_3.add(taSSocial);
subpanel_3.add(Box.createRigidArea(new Dimension(5,0)));
subpanel_3.add(taSSCampo);
panel_intermedio.add(subpanel_3);

JPanel subpanel_4 = new JPanel();
```

```
        subpanel_4.add(fecNac);
        subpanel_4.add(Box.createRigidArea(new Dimension(5,0)));
        subpanel_4.add(fecNac1Campo);
        JLabel etiq1 = new JLabel("/");
        subpanel_4.add(etiq1);
        subpanel_4.add(fecNac2Campo);
        JLabel etiq2 = new JLabel("/");
        subpanel_4.add(etiq2);
        subpanel_4.add(fecNac3Campo);
        JLabel aptoEtiqueta =new JLabel("¿Es apto el paciente para CmA?");
        subpanel_4.add(aptOEtiqueta);
        subpanel_4.add(Box.createRigidArea(new Dimension(10,0)));
        subpanel_4.add(si);
        subpanel_4.add(Box.createRigidArea(new Dimension(10,0)));
        subpanel_4.add(no);

        panel_intermedio.add(subpanel_4);

        panel_intermedio.setPreferredSize(new Dimension(750,90));
        add(panel_intermedio);
    }
}
```

Clase PanelDatosClinicos.java

```
/**
 *   Panel para los datos clínicos del paciente
 */

package huvr.paneles;

import javax.swing.*;
import java.awt.*;
import java.io.*;
import java.net.*;
import huvr.paneles.*;
import huvr.nucleo.FuncionesGUI;
import huvr.escritores.*;

public class PanelDatosClinicos extends javax.swing.JPanel
{
    // Ponemos los componentes como variables públicas para su acceso
    // exterior. Iremos agrupando áreas de texto + panel de scroll

    public static JTextArea cuidadosPreoperatorios;
    public JScrollPane cuidPreopScrollPane;
    public static JTextArea noAptoCmA;
    public JScrollPane noAptoScrollPane;
    public static JTextField fecha1Campo;
    public static JTextField fecha2Campo;
    public static JTextField fecha3Campo;
    public JLabel fecha;
    public static JTextField hora1Campo;
    public static JTextField hora2Campo;
    public JLabel hora;
    public JButton enviar;
    public JButton limpiarDatos;
    public JButton cancelar;
    /** Oyentes para los botones */
    OyenteEventoEnviar oyente1 = new OyenteEventoEnviar();
    OyenteEventoLimpiar oyente2 = new OyenteEventoLimpiar();
}
```

```
OyenteEventoCancelar oyente3 = new OyenteEventoCancelar();

/**
 * Constructor de la clase
 */
public PanelDatosClinicos() {

    cuidadosPreoperatorios = new JTextArea();
    cuidadosPreoperatorios.setToolTipText("Escriba los cuidados
    preoperatorios que debe seguir el paciente");
    // habilitamos el área de texto y dejamos editar
    cuidadosPreoperatorios.setEnabled(true);
    cuidadosPreoperatorios.setEditable(true);
    // configuramos que salte de línea si no cabe, sin cortar las
    // palabras
    cuidadosPreoperatorios.setLineWrap(true);
    cuidadosPreoperatorios.setWrapStyleWord(true);
    cuidadosPreoperatorios.setBorder(BorderFactory.createLineBorder
    (Color.gray));
    // incrustamos en un panel de scroll
    cuidPreopScrollPane = new JScrollPane(cuidadosPreoperatorios);
    // configuramos barra de desplazamiento vertical
    cuidPreopScrollPane.setVerticalScrollBarPolicy(
    JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED);

    noAptoCmA = new JTextArea();
    noAptoCmA.setToolTipText("Indique los motivos por los que el
    paciente no es apto para CmA");
    noAptoCmA.setEnabled(true);
    noAptoCmA.setEditable(true);
    noAptoCmA.setLineWrap(true);
    noAptoCmA.setWrapStyleWord(true);
    noAptoCmA.setBorder(BorderFactory.createLineBorder(Color.gray));
    noAptoScrollPane = new JScrollPane(noAptoCmA);
    noAptoScrollPane.setVerticalScrollBarPolicy(
    JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED);

    fecha1Campo = new JTextField("", 2);
    fecha2Campo = new JTextField("", 2);
    fecha3Campo = new JTextField("", 4);
    fecha1Campo.setToolTipText("Inserte la fecha de la intervención,
    ej: 05/07/2004");
    fecha2Campo.setToolTipText("Inserte la fecha de la intervención,
    ej: 05/07/2004");
    fecha3Campo.setToolTipText("Inserte la fecha de la intervención,
    ej: 05/07/2004");
    fecha = new JLabel("Fecha prevista: ");
    fecha.setToolTipText("Inserte la fecha de la intervención,
    ej: 05/07/2004");
    fecha.setLabelFor(fecha1Campo);

    hora1Campo = new JTextField("", 2);
    hora2Campo = new JTextField("", 2);
    hora1Campo.setToolTipText("Inserte la hora de la intervención,
    ej: 17:30");
    hora2Campo.setToolTipText("Inserte la hora de la intervención,
    ej: 17:30");
    hora = new JLabel("Hora: ");
    hora.setToolTipText("Inserte la hora de la intervención,
    ej: 17:30");
    hora.setLabelFor(hora1Campo);
```

```
enviar = new JButton("Enviar");
limpiarDatos = new JButton("Limpiar datos");
cancelar = new JButton("Cancelar");
// Configuramos el acceso directo a los botones
enviar.setMnemonic(event.KeyEvent.VK_E);
enviar.addActionListener(oyente1);
limpiarDatos.setMnemonic(event.KeyEvent.VK_L);
limpiarDatos.addActionListener(oyente2);
cancelar.setMnemonic(event.KeyEvent.VK_C);
cancelar.addActionListener(oyente3);

// A continuación, colocamos los componentes en su sitio

setLayout(new BorderLayout(this, BorderLayout.Y_AXIS));

JPanel subpanel1 = new JPanel();
subpanel1.setLayout(new GridLayout(1,2));
cuidPreopScrollPane.setBorder(BorderFactory.
    createTitledBorder("Cuidados Preoperatorios"));
subpanel1.add(cuidPreopScrollPane);
noAptoScrollPane.setBorder(BorderFactory.createTitledBorder
    ("Motivos por los que no es apto para CmA"));
subpanel1.add(noAptoScrollPane);
subpanel1.setPreferredSize(new Dimension(750,90));
add(subpanel1);

JPanel subpanel2 = new JPanel();
subpanel2.add(fecha);
subpanel2.add(Box.createRigidArea(new Dimension(5,0)));
subpanel2.add(fecha1Campo);
JLabel aux1 = new JLabel("/");
subpanel2.add(aux1);
subpanel2.add(fecha2Campo);
JLabel aux2 = new JLabel("/");
subpanel2.add(aux2);
subpanel2.add(fecha3Campo);
subpanel2.add(Box.createRigidArea(new Dimension(10,0)));
subpanel2.add(hora);
subpanel2.add(Box.createRigidArea(new Dimension(5,0)));
subpanel2.add(hora1Campo);
JLabel aux3 = new JLabel(":");
subpanel2.add(aux3);
subpanel2.add(hora2Campo);
subpanel2.setBorder(BorderFactory.createTitledBorder
    ("Fecha y hora de la intervención"));
subpanel2.setPreferredSize(new Dimension(750,30));
add(subpanel2);

JPanel subpanel4 = new JPanel();
subpanel4.add(enviar);
subpanel4.add(limpiarDatos);
subpanel4.add(cancelar);
add(subpanel4);

setPreferredSize(new Dimension(750,220));

}

class OyenteEventoEnviar implements event.ActionListener{
    public void actionPerformed (event.ActionEvent e){
```

```
boolean f = new FuncionesGUI().creaArchivo();
if (f == true){
/**
 * Aquí va el código destinado a enviar el archivo que hemos
 * creado desde la central del HUVR hasta la Gestoría de Usuarios
 */
    try{
        /**
         * Primero, crea un objeto de tipo URL para conectarse con
         * el servidor ftp. El argumento que se le pasa a la clase
         * será la referencia absoluta a dicho servidor
         */
        URL url = new URL("ftp://ftp.servidor_gestoria.sas");
        // A continuación, establece la conexión
        URLConnection urlc = url.openConnection();
        // Obtiene los canales de entrada y salida de datos
        InputStream is = urlc.getInputStream();
        OutputStream os = urlc.getOutputStream();
        // Envía el fichero mediante la clase enviaFich
        enviaFich fich = new enviaFich(os);
        // Ya solo nos queda cerrar la conexión
        is.close();
        os.close();

    }
    catch (Throwable t) {
        new FuncionesGUI().informar("actionPerformed(): " + new
            escritorError().escribirError(t));
    }

}
else
    JOptionPane.showMessageDialog(null,"No se pudo enviar el
        archivo porque faltaba alguno de los campos");
}

}

public class OyenteEventoLimpiar implements event.ActionListener{
    public void actionPerformed (event.ActionEvent e){
        PanelDatosPersonales.fecNac1Campo.setText("");
        PanelDatosPersonales.fecNac2Campo.setText("");
        PanelDatosPersonales.fecNac3Campo.setText("");
        PanelDatosPersonales.afiliSSCampo.setText("");
        PanelDatosPersonales.apell1Campo.setText("");
        PanelDatosPersonales.apell2Campo.setText("");
        PanelDatosPersonales.domicilioCampo.setText("");
        PanelDatosPersonales.localidadCampo.setText("");
        PanelDatosPersonales.nombreCampo.setText("");
        PanelDatosPersonales.taSSCampo.setText("");
        PanelDatosPersonales.tfnoCampo.setText("");

        PanelDatosClinicos.cuidadosPreoperatorios.setText("");
        PanelDatosClinicos.fecha1Campo.setText("");
        PanelDatosClinicos.fecha2Campo.setText("");
        PanelDatosClinicos.fecha3Campo.setText("");
        PanelDatosClinicos.hora1Campo.setText("");
        PanelDatosClinicos.hora2Campo.setText("");
        PanelDatosClinicos.noAptoCmA.setText("");
    }
}
```

```
class OyenteEventoCancelar implements event.ActionListener{
    public void actionPerformed(event.ActionEvent e) {
        // Este oyente simplemente hace que salgamos de la aplicación
        System.exit(0);
    }
}

class enviaFich implements Serializable{
    public enviaFich(OutputStream os){
        try{
            // Asocia el ObjectOutputStream al OutputStream por el que se
            // enviará el fichero
            ObjectOutputStream objout = new ObjectOutputStream(os);
            objout.writeObject("c:/ficheros/" + FuncionesGUI.nombreFich +
                ".txt");
            objout.close();
        }
        catch(Throwable t){
            new FuncionesGUI().informar(new
                escritorError().escribirError(t));
        }
    }
}
```

El resto de clases están contenidas en los paquetes *huvr.nucleo*, *huvr.escritores* y *huvr.menus*; como son muy parecidas a las mismas clases en las otras dos aplicaciones, no incluiremos el código aquí, aunque sí que está en el CD correspondiente.