

PRACTICA DE LENGUAJES DE PROGRAMACIÓN

JUNIO 2006

Alumno: Victor Martín del Canto Godino

Correo: vdc61@usuarios.retecal.es

Tfno: 983211989

Dejando aun lado el interfaz grafico que es necesario crear, y centrándonos en el problema en si, se trata de tomar los elementos de un texto, las palabras, y realizar un tratamiento con ellas. Estas palabras se dividen en tres subconjuntos bien definidos, excepciones, conocidas y desconocidas. Ahora bien es necesario hacer algunas precisiones.

Una palabra puede aparecer en el texto original cero, una o mas ocasiones. Para saber si esa palabra pertenece a uno de los subconjuntos no es necesario mas que aparezca una vez en dicho subconjunto, no todas las apariciones que tiene en el texto.

En segundo lugar una palabra no puede aparecer en mas de un subconjunto, solo puede estar en uno de los tres.

En tercer lugar hay que tener una relación de la situación de cada palabra en el texto a tratar y su pertenencia a cada subconjunto.

Por tanto he creado la siguiente estructura de datos

Conjunto (TreeSet) de Excepciones.

Conjunto (TreeSet) de Conocidos.

Conjunto (TreeSet) de Desconocidos.

Elegidos por su característica de que cada elemento de un conjunto es único, y cada elemento que se añade queda ordenado alfabéticamente.

Lista (ArrayList) de Palabras del Texto original.

Lista (LinkedList) de posiciones en el Texto de palabras que son Excepciones.

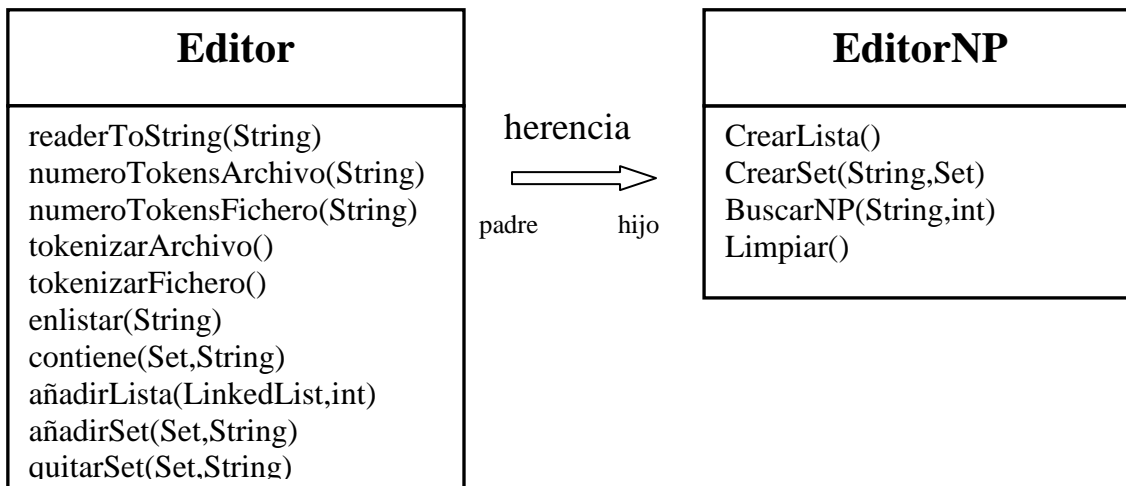
Lista (LinkedList) de posiciones en el Texto de palabras que son Conocidos.

Lista (LinkedList) de posiciones en el Texto de palabras que son Desconocidos.

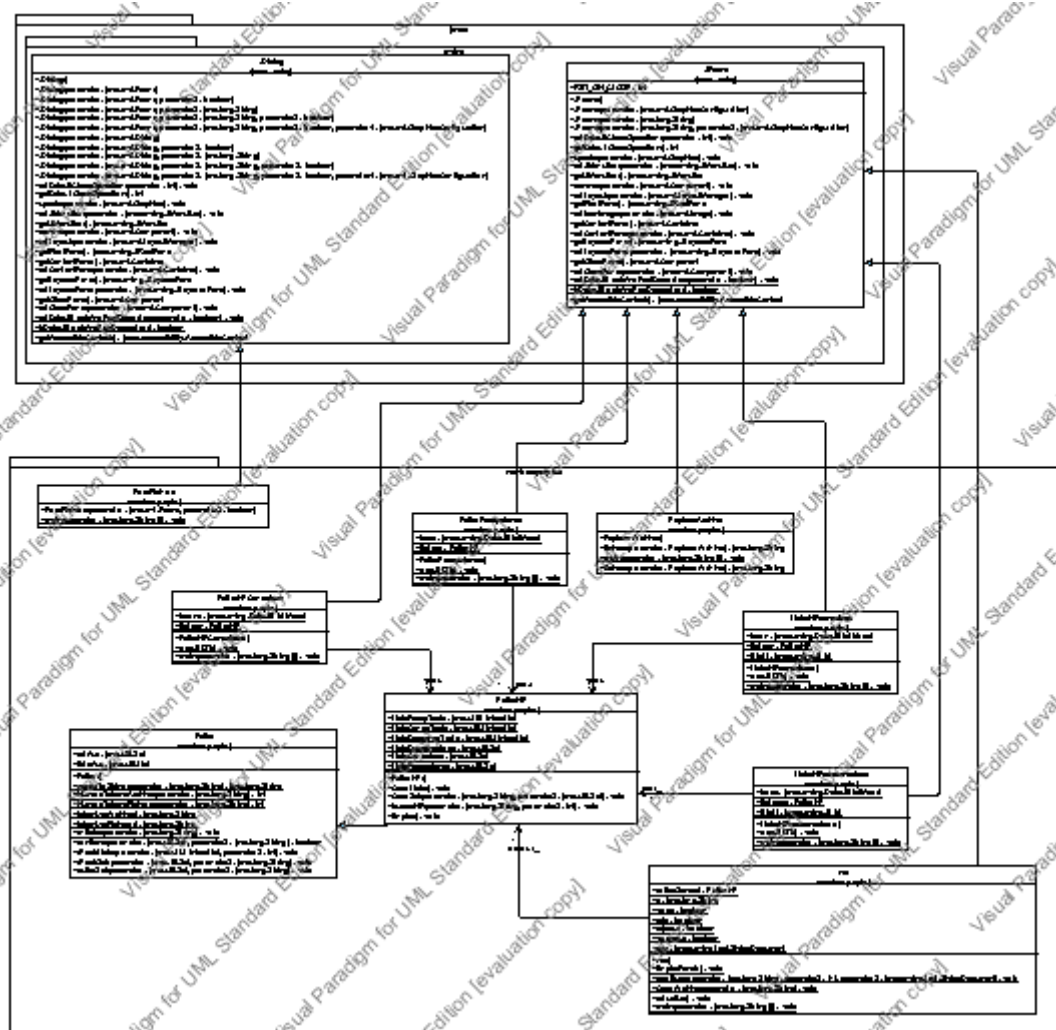
Todo el texto a analizar es descompuesto en tokens (palabras y delimitadores) y se almacena en un ArrayList. A continuación se analiza cada palabra su inclusión en el subconjunto que le corresponda; por defecto si no es Excepción ni Conocido, pasa a pertenecer a Desconocido.

Al mismo tiempo se incluye en la Lista (LinkedList) que corresponde la posición que esa palabra ocupa en el Texto (se toman en cuenta la posición de los delimitadores, presentes en la ArrayList, pero ausentes en los conjuntos (TreeSet)).

Teniendo en cuenta estas consideraciones y como he dicho dejando a un lado el interfaz grafico tengo el siguiente diagrama de clases.



A continuación presento tres formas de representar el diagrama de clases (se adjuntan los archivos graficos en el cd para poderlo ver en profundidad y con nitidez)



```
public class ner extends javax.swing.JFrame
```

Clase principal, ventana donde se lanza a las demas, segun enunciado de la practica

```
public static void limpiarPanel() throws BadLocationException {
```

Si el panel contiene texto se borra todo

```
public static void escritura(String cadena,int version,StyledDocument doc) throws  
BadLocationException {
```

Escribe los caracteres en el color adecuado

```
public static void CrearArchivo(String nombre) throws IOException {  
    BufferedReader arch=new BufferedReader (new FileReader (nombre));
```

Se abre un archivo, si no existia se crea nuevo

```
public void actualiza() {
```

Actualiza el contenido del panel

```
public class Editor
```

Herramientas generales para el tratamiento del texto

```
public static String readerToString (String f)throws IOException
```

Extrae el contenido del archivo en una cadena

```
public static int NumeroTokensArchivo(String str) throws IOException
```

Numero de tokens del archivo a tratar (incluyendo delimitadores)

```
public static int NumeroTokensFichero(String str) throws IOException
```

Numero de tokens del fichero(Excepciones.txt o Conocidos.txt) a tratar (excluyendo delimitadores)

```
public static String tokenizarArchivo ()
```

Devuelve los tokens (incluyendo delimitadores) del archivo

```
public static String tokenizarFichero ()
```

Devuelve los tokens (excluyendo delimitadores) del fichero

```
public static void enlistar (String cadena) throws IOException
```

Añade la cadena al ArrayList

```
public boolean contiene(Set setAux ,String cadena)
```

Devuelve cierto si la cadena esta en el conjunto


```
public void añadirLista(LinkedList listaAux, int indice
```

Añade la posicion entera a la LinkedList y ordena la lista

```
public void añadirSet(Set setAux,String cadena)
```

Añade la cadena al conjunto

```
public void quitarSet(Set setAux,String cadena)
```

Quita la cadena del conjunto

```
public class EditorNP extends Editor
```

Tratamiento de un texto para obtener NOMBRES PROPIOS (segun practica)

```
public void CrearLista()throws IOException
```

Analiza cada token del archivo busca si es Nombre Propio. Si lo es buscamos si esta en Excepciones o Conocidos y se añade su posicion al LinkedList que corresponda. Colocamos cada token en el ArrayList en su posicion

```
public void CrearSet(String fichero,Set setAux) throws IOException
```

Analiza cada fichero y añade cada token al conjunto correspondiente

```
public void buscarNP(String cadena,int indice)
```

Busca cada cadena en los conjuntos añade su posicion al LinkedList que corresponda

```
public void limpiar()
```

Limpia de elementos todas las listas

```
public class EditarExcepciones extends javax.swing.JFrame
```

Edita las Excepciones, pudiendo añadir o eliminar elementos

```
public static void creaJLIST()
```

Pone los elemetos del conjunto Excepciones en el Jlist

```
public class EditarNPConocidos extends javax.swing.JFrame
```

Edita los NP Conocidos, pudiendo añadir o eliminar elementos

```
public static void creaJLIST()
```

Pone los elementos del conjunto Excepciones en el Jlist

```
public class ExplorarArchivo extends javax.swing.JFrame
```

Explorador de archivos para seleccionar el archivo a tratar

```
public String archivo(ExplorarArchivo explordialog)
```

Devuelve el nombre de archivo seleccionado y su trayectoria

```
public class ListarNPconocidos extends javax.swing.JFrame
```

Lista los NP Conocidos que hay en el texto

```
public static void creaJLIST()
```

Pone los elemetos del conjunto Excepciones en el Jlist

```
public class ListarNPdesconocidos extends javax.swing.JFrame
```

Lista los NP Desconocidos que hay en el texto pudiendo modificar cada elemento para que sea Excepcion o Conocido

```
public static void creaJLIST()
```

Pone los elemetos del conjunto Excepciones en el Jlist

```
public class ErrorFichero extends javax.swing.JDialog
```

Ventana de error cuando el nombre del archivo o su trayectoria son incorrectos

CODIGO

C:\WINDOWS\NombresPropios\src\nombrespropios\ner.java

```
/*
 * ner.java
 *
 * Created on 3 de abril de 2006, 18:15
 */

package nombrespropios;

import javax.swing.*;
import java.awt.*;
import java.io.*;
import java.util.*;
import javax.swing.text.*;
/**
 *
 * @author Victor del Canto
 */
public class ner extends javax.swing.JFrame {

    public static EditorNP editorGeneral=new EditorNP();
    public static String g= new String();
    public static boolean verde=false;
    public static boolean rojo=false;
    public static boolean rojoaux=false;
    public static boolean verdeaux=false;
    public static StyledDocument doc;
    /** Creates new form ner */
    public ner() {
        initComponents();
        this.setSize(782,500);
    }

    /** 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.
     */
    // <editor-fold defaultstate="collapsed" desc="Generated Code ">
    private void initComponents() {
        java.awt.GridBagConstraints gridBagConstraints;

        Explorar = new javax.swing.JPopupMenu();
        editarexcepciones = new javax.swing.JButton();
        editarnpconocidos = new javax.swing.JButton();
        fichero = new javax.swing.JTextField();
        jLabel2 = new javax.swing.JLabel();
        Explora = new javax.swing.JButton();
        abrir = new javax.swing.JButton();
        jScrollPane2 = new javax.swing.JScrollPane();
        jTextPane1 = new javax.swing.JTextPane();
        marcardesconocidos = new javax.swing.JCheckBox();
        marcarconocidos = new javax.swing.JCheckBox();
        jLabel1 = new javax.swing.JLabel();
        listarconocidos = new javax.swing.JButton();
    }
}
```

```

listardesconocidos = new javax.swing.JButton();

Explorar.setInvoker(Explora);

getContentPane().setLayout(new java.awt.GridBagLayout());

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("programa");
addWindowFocusListener(new java.awt.event.WindowFocusListener() {
    public void windowGainedFocus(java.awt.event.WindowEvent evt) {
        formWindowGainedFocus(evt);
    }
    public void windowLostFocus(java.awt.event.WindowEvent evt) {
    }
});
addWindowListener(new java.awt.event.WindowAdapter() {
    public void windowActivated(java.awt.event.WindowEvent evt) {
        formWindowActivated(evt);
    }
});

editarexcepciones.setText("Editar Excepciones");
editarexcepciones.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        editarexcepcionesMouseClicked(evt);
    }
});

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 1;
gridBagConstraints.gridy = 1;
gridBagConstraints.gridwidth = 2;
getContentPane().add(editarexcepciones, gridBagConstraints);

editarnpconocidos.setText("Editar NP Conocidos");
editarnpconocidos.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        editarnpconocidosMouseClicked(evt);
    }
});

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 3;
gridBagConstraints.gridy = 1;
gridBagConstraints.gridwidth = 2;
getContentPane().add(editarnpconocidos, gridBagConstraints);

fichero.setColumns(30);
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 3;
gridBagConstraints.gridy = 4;
gridBagConstraints.gridwidth = 2;
gridBagConstraints.fill = java.awt.GridBagConstraints.BOTH;
getContentPane().add(fichero, gridBagConstraints);

jLabel2.setText("Fichero:");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 2;
gridBagConstraints.gridy = 4;
getContentPane().add(jLabel2, gridBagConstraints);

Explora.setText("Explorar");

```

```

Explora.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        ExploraMouseClicked(evt);
    }
});

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 5;
gridBagConstraints.gridy = 4;
getContentPane().add(Explora, gridBagConstraints);

abrir.setText("Abrir");
abrir.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        abrirMouseClicked(evt);
    }
});

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 6;
gridBagConstraints.gridy = 4;
getContentPane().add(abrir, gridBagConstraints);

jScrollPane2.setBackground(new java.awt.Color(255, 255, 255));
jScrollPane2.setBorder(new javax.swing.border.EtchedBorder());

jScrollPane2.setVerticalScrollBarPolicy(javax.swing.ScrollPaneConstants.VERTICAL_SCROLL
BAR_ALWAYS);
jScrollPane2.setAutoscrolls(true);
jScrollPane2.setMaximumSize(new java.awt.Dimension(625, 550));
jScrollPane2.setMinimumSize(new java.awt.Dimension(175, 150));
jScrollPane2.setPreferredSize(new java.awt.Dimension(225, 250));
jTextPanel1.setDocument(jTextPanel1.getStyledDocument());
jScrollPane2.setViewportView(jTextPanel1);

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 6;
gridBagConstraints.gridwidth = java.awt.GridBagConstraints.REMAINDER;
gridBagConstraints.gridheight = 10;
gridBagConstraints.fill = java.awt.GridBagConstraints.BOTH;
gridBagConstraints.ipadx = 2;
gridBagConstraints.ipady = 2;
gridBagConstraints.insets = new java.awt.Insets(0, 1, 0, 1);
getContentPane().add(jScrollPane2, gridBagConstraints);

marcardesconocidos.setText("Marcar NP Desconocidos (rojo)");
marcardesconocidos.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        marcardesconocidosMouseClicked(evt);
    }
});

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 4;
gridBagConstraints.gridy = 18;
getContentPane().add(marcardesconocidos, gridBagConstraints);

marcarconocidos.setText("Marcar NP Conocidos (verde)");
marcarconocidos.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {

```



```

        marcarconocidosMouseClicked (evt) ;
    }
});

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 2;
gridBagConstraints.gridy = 18;
getContentPane().add(marcarconocidos, gridBagConstraints);

jLabel1.setText("Listar Np reconocidos:");
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 2;
gridBagConstraints.gridy = 20;
getContentPane().add(jLabel1, gridBagConstraints);

listarconocidos.setText("Conocidos");
listarconocidos.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        listarconocidosMouseClicked(evt);
    }
});

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 3;
gridBagConstraints.gridy = 20;
getContentPane().add(listarconocidos, gridBagConstraints);

listardesconocidos.setText("ListarNPDesconocidos");
listardesconocidos.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        listardesconocidosMouseClicked(evt);
    }
});

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 4;
gridBagConstraints.gridy = 20;
getContentPane().add(listardesconocidos, gridBagConstraints);

setBounds(120, 90, 400, 300);
}
// </editor-fold>

private void formWindowGainedFocus(java.awt.event.WindowEvent evt) {
    actualiza();
}

private void formWindowActivated(java.awt.event.WindowEvent evt) {
    actualiza();
}

private void marcarconocidosMouseClicked(java.awt.event.MouseEvent evt) {
    // Con la pulsacion del boton izquierdo del raton cambia el estado de verde
    (booleano)
    if ((evt.getModifiers() & java.awt.event.InputEvent.BUTTON1_MASK) != 0) {
        /*
        if (g.equals(null)==false){ //existe un archivo que abrir */
        verde=!verde;
        actualiza(); //actualiza el texto en la ventana
    }
}

private void marcadesconocidosMouseClicked(java.awt.event.MouseEvent evt) {

```

```

// Con la pulsacion del boton izquierdo del raton cambia el estado de rojo
(booleano)
    if ((evt.getModifiers() & java.awt.event.InputEvent.BUTTON1_MASK) != 0) {
/*
        if (g.equals(null)==false){ //existe un archivo que abrir*/
            rojo=!rojo;
            actualiza(); //actualiza el texto en la ventana
        }
    }

private void abrirMouseClicked(java.awt.event.MouseEvent evt) {
    g=fichero.getText();//toma el nombre del archivo seleccionado
    actualiza(); // actualiza el panel (metodo publico)
}

private void listardesconocidosMouseClicked(java.awt.event.MouseEvent evt) {
    ListarNPdesconocidos lisdes=new ListarNPdesconocidos(); //Se crea el objeto
    lisdes.lsdes.clear(); //Inicializacion
    lisdes.creaJLIST(); //y
    lisdes.setVisible(true); //activacion
}

private void listarconocidosMouseClicked(java.awt.event.MouseEvent evt) {
    ListarNPconocidos liscon=new ListarNPconocidos(); //Se crea el objeto
    liscon.lscon.clear(); //Inicializacion
    liscon.creaJLIST(); //y
    liscon.setVisible(true); //activacion
}

private void editarnpconocidosMouseClicked(java.awt.event.MouseEvent evt) {
    EditarNPConocidos edcon= new EditarNPConocidos(); //Se crea el objeto
    edcon.lscono.clear(); //Inicializacion
    edcon.creaJLIST(); //y
    edcon.setVisible(true); //activacion
}

private void ExploraMouseClicked(java.awt.event.MouseEvent evt) {
    ExplorarArchivo explordialog = new ExplorarArchivo(); //Se crea el objeto
    explordialog.setVisible(true); //activacion
    String sg=explordialog.fichero(explordialog); //fichero escogido
    fichero.setText(sg); //se muestra en la pantalla
    explordialog.setVisible(false); //ocultacion
    remove(explordialog); //y cierre
}

private void editarexcepcionesMouseClicked(java.awt.event.MouseEvent evt) {
    EditarExcepciones edex= new EditarExcepciones(); //Se crea el objeto
    edex.lsexcc.clear(); //Inicializacion
    edex.creaJLIST(); //y
    edex.setVisible(true); //activacion
}

public static void limpiarPanel() throws BadLocationException {
    if (doc.equals(null)==false) { // Si el panel contiene texto
        doc.remove(0,doc.getLength()); //se borra todo
        //escritura(" ",3,doc);
    }
}

public static void escritura(String cadena,int version,StyledDocument doc) throws
BadLocationException {

```

```

//para el rojo
    Style estiloRojo = jTextPanel.addStyle (null, null) ;
    StyleConstants.setForeground (estiloRojo, Color.red) ;
//para el verde
    Style estiloVerde = jTextPanel.addStyle (null, null) ;
    StyleConstants.setForeground (estiloVerde, Color.green) ;
//para el negro
    Style estiloNegro = jTextPanel.addStyle (null, null) ;
    StyleConstants.setForeground (estiloNegro, Color.black) ;

    switch (version) {
        case 1:
            doc.insertString (doc.getLength () , cadena, estiloVerde) ;
            break;
        case 2:
            doc.insertString (doc.getLength () , cadena, estiloRojo) ;
            break;
        case 3:
            doc.insertString (doc.getLength () , cadena, estiloNegro) ;
            break;
        default:
            doc.insertString (doc.getLength () , cadena, estiloNegro) ;
    }
}

public static void CrearArchivo (String nombre) throws IOException {
    BufferedReader arch=new BufferedReader (new FileReader (nombre)) ; //Se abre un
archivo
    if (arch.read ()== -1) { //si no existia
        new File (nombre) .createNewFile () ; //se crea nuevo
    }
}

public void actualiza () {
    try {
        doc=jTextPanel.getStyledDocument () ; //estilo de escritura rojo, verde o
negro
        limpiarPanel () ; // se borra el panel (para actualizaciones)
        if (g.equals ("") ==false) { //existe un archivo que abrir
            editorGeneral.limpiar () ;
            editorGeneral.CrearLista () ;
            int elem=editorGeneral.lista.size () ; // numero de tokens del
archivo
            for (int x=0; x<(elem); x++) {
                // Se escribe en el panel cada token segun le corresponda este o
no activada su marca
                if
((editorGeneral.ListaConocTexto.contains (x) ==true) && (verde==true)) {
                    escritura (editorGeneral.lista.get (x) .toString () , 1, doc) ;
                } else if
((editorGeneral.ListaDesconocTexto.contains (x) ==true) && (rojo==true)) {
                    escritura (editorGeneral.lista.get (x) .toString () , 2, doc) ;
                } else { escritura (editorGeneral.lista.get (x) .toString () , 3, doc) ;
                }
            }
        } else {
            editorGeneral.limpiar () ;
        }
    }
}

```



```

    } catch (IOException e) {
        JFrame err=new JFrame();
        ErrorFichero error=new ErrorFichero(err, true);
        error.setVisible(true);
    } catch (BadLocationException e){
        System.out.println("error BadLocation");
    }
}
/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                // Toma la apariencia del sistema
                UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
                //Se crean los archivos Excepciones.txt y Conocidos.txt si no
                CrearArchivo("Excepciones.txt");
                CrearArchivo("Conocidos.txt");
                //Se añaden los contenidos de Excpcion.txt y Conocidos.txt a sus
                conjuntos correspondientes

                editorGeneral.CrearSet("Excepciones.txt",editorGeneral.ListaExcepciones);

                editorGeneral.CrearSet("Conocidos.txt",editorGeneral.ListaConocidos);
            } catch (Exception e) {
            }

            new ner().setVisible(true);

        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton Explora;
private javax.swing.JPopupMenu Explorar;
private javax.swing.JButton abrir;
private javax.swing.JButton editarexcepciones;
private javax.swing.JButton editarnpconocidos;
private javax.swing.JTextField fichero;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JScrollPane jScrollPane2;
private static javax.swing.JTextPane jTextPanel1;
private javax.swing.JButton listarconocidos;
private javax.swing.JButton listardesconocidos;
private javax.swing.JCheckBox marcarconocidos;
private javax.swing.JCheckBox marcardesconocidos;
// End of variables declaration
}

```

```

/*
 * Editor.java
 *
 * Created on 10 de abril de 2006, 11:51
 *
 * To change this template, choose Tools | Options and locate the template under
 * the Source Creation and Management node. Right-click the template and choose
 * Open. You can then make changes to the template in the Source Editor.
 */

package nombrespropios;

import java.io.*;
import java.util.*;
/**
 *
 * @author Victor del Canto
 */
public class Editor {           //Herramientas generales para el tratamiento del texto

    public Set setAux = new TreeSet();
    public List listaAux= new LinkedList();
    static int num;
    static ArrayList lista = new ArrayList();
    static StringTokenizer stA,stF;

    /** Creates a new instance of Editor */
    public Editor() {
    }

    protected static final int BLKSIZ=8192;

    public static String readerToString (String f)throws IOException {
    // Extrae el contenido del archivo en una cadena
        FileReader is=new FileReader(f);
        StringBuffer sb= new StringBuffer();
        char[]b=new char[BLKSIZ];
        int n;

        //Lee un bloque. Si obtiene algun caracter, los añade.
        while ((n=is.read(b))>0) {
            sb.append(b,0,n);
        }
        return sb.toString();
    }

    public static int NumeroTokensArchivo(String str)throws IOException{
    //Numero de tokens del archivo a tratar (incluyendo delimitadores)
        stA=new StringTokenizer(readerToString(str), "\n"+"\r" " "+"."+"/"+"*"+"-
"+"+"+" ("+"") "+"{"+"}"+"["++"]"+"_"+" ":"+" ";"+" ","+"?"+";"+"¿"+"!"+"#"+"="+" " ,true);
        num=stA.countTokens();
        return num;
    }

    public static int NumeroTokensFichero(String str)throws IOException{

```

```

//Numero de tokens del fichero(Excepciones.txt o Conocidos.txt) a tratar
(excluyendo delimitadores)
    stF=new StringTokenizer(readerToString(str), "\n"+"\r" " "+"["+"+"+", "+", false);
    num=stF.countTokens();
    return num;
}

public static String tokenizarArchivo () {
//Devuelve los tokens (incluyendo delimitadores) del archivo
    while (stA.hasMoreTokens()) {
        return stA.nextToken();
    }
    return (" ");
}

public static String tokenizarFichero () {
//Devuelve los tokens (excluyendo delimitadores) del fichero
    while (stF.hasMoreTokens()) {
        return stF.nextToken();
    }
    return (" ");
}

public static void enlistar (String cadena) throws IOException{
//Añade la cadena al ArrayList
    lista.add(cadena);
}

public boolean contiene(Set setAux ,String cadena) {
//Devuelve cierto si la cadena esta en el conjunto
    return setAux.contains(cadena);
}

public void añadirLista(LinkedList listaAux, int indice) {
//Añade la posicion entera a la LinkedList y ordena la lista
    listaAux.addLast(indice);
    Collections.sort(listaAux);
}

public void añadirSet(Set setAux,String cadena) {
//Añade la cadena al conjunto
    setAux.add(cadena);
}

public void quitarSet(Set setAux,String cadena) {
// Quita la cadena del conjunto
    setAux.remove(cadena);
}
}

```



```

/*
 * EditorNP.java
 *
 * Created on 13 de abril de 2006, 18:18
 *
 * To change this template, choose Tools | Options and locate the template under
 * the Source Creation and Management node. Right-click the template and choose
 * Open. You can then make changes to the template in the Source Editor.
 */

package nombrespropios;

import java.io.*;
import java.util.*;

/**
 *
 * @author Victor del Canto
 */
public class EditorNP extends Editor{
//Tratamiento de un texto para obtener NOMBRES PROPIOS (segun practica)
    public static LinkedList ListaExcepTexto = new LinkedList();
    public static LinkedList ListaConocTexto = new LinkedList();
    public static LinkedList ListaDesconocTexto = new LinkedList();
    public static Set ListaDesconocidos = new TreeSet();
    public static Set ListaConocidos = new TreeSet();
    public static Set ListaExcepciones = new TreeSet();
    /** Creates a new instance of EditorNP */
    public EditorNP() {
    }

    public void CrearLista() throws IOException {
limpiar();
num=    NumeroTokensArchivo(ner.g);
for (int c=0;c<num;c++){    //Analiza cada token del archivo
    String cadena=tokenizarArchivo();    //Busca
    String inicial=cadena.substring(0,1);    // si es
    String cadenaMAY=cadena.toLowerCase();    //Nombre
    String INICMAY=cadenaMAY.substring(0,1);    //Propio
    if (inicial.equals(INICMAY)==false) {    //Si lo es
//buscamos si esta en Excepciones o Conocidos
//y se a ade su posicion al LinkedList que corresponda
        buscarNP(cadena,c);    //
    }
    // Colocamos cada token en el ArrayList en su posicion
    enlistar(cadena);
}
}

    public void CrearSet(String fichero,Set setAux) throws IOException {
//Analiza cada fichero y a ade cada token al conjunto correspondiente
num=    NumeroTokensFichero(fichero);
for (int c=0;c<num;c++){
    String cadena=tokenizarFichero();
    a adirSet(setAux,cadena);
}
}
}

```

```

public void buscarNP(String cadena,int indice) {
    // Busca cada cadena en los conjuntos
    //y añade su posicion al LinkedList que corresponda
    if (contiene(ListaExcepciones,cadena)==true) {
        añadirLista(ListaExcepTexto,indice);
    }else if (contiene(ListaConocidos,cadena)==true) {
        añadirLista(ListaConocTexto,indice);
    }else {
        ListaDesconocidos.add(cadena);
        añadirLista(ListaDesconocTexto,indice);
    }
}

public void limpiar() {
    //Limpia de elementos todas las listas
    ListaExcepTexto.clear();
    ListaConocTexto.clear();
    ListaDesconocTexto.clear();
    lista.clear();
}
}

```

```

/*
 * EditarExcepciones.java
 *
 * Created on 9 de abril de 2006, 19:09
 */

package nombrespropios;
import javax.swing.*.*;
import java.util.*;
import java.io.*;
/**
 *
 * @author Victor del Canto
 */
public class EditarExcepciones extends javax.swing.JFrame {

    public static DefaultListModel lsexcc=new DefaultListModel();
    public static EditorNP jlistexc= new EditorNP();
    /** Creates new form EditarExcepciones */
    public EditarExcepciones() {
        initComponents();
    }

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

        añadir = new javax.swing.JButton();
        eliminar = new javax.swing.JButton();
        cerrar = new javax.swing.JButton();
        jScrollPane1 = new javax.swing.JScrollPane();
        jList1 = new javax.swing.JList();
        Excepcion = new javax.swing.JTextField();
        jLabel1 = new javax.swing.JLabel();
        error = new javax.swing.JLabel();

        getContentPane().setLayout(new java.awt.GridBagLayout());

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("EDITOR DE EXCEPCIONES");
        setFocusCycleRoot(false);
        setFocusTraversalPolicy(null);
        setFocusable(false);
        añadir.setText("A\u00f1adir");
        añadir.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                añadirMouseClicked(evt);
            }
        });

        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 5;
        gridBagConstraints.gridy = 10;
        gridBagConstraints.gridheight = 2;
        gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
        gridBagConstraints.insets = new java.awt.Insets(20, 10, 38, 0);
    }

```



```

getContentPane().add(añadir, gridBagConstraints);

eliminar.setText("Eliminar");
eliminar.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        eliminarMouseClicked(evt);
    }
});

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 9;
gridBagConstraints.gridy = 0;
gridBagConstraints.gridwidth = 2;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
gridBagConstraints.insets = new java.awt.Insets(37, 11, 3, 35);
getContentPane().add(eliminar, gridBagConstraints);

cerrar.setText("Cerrar");
cerrar.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        cerrarMouseClicked(evt);
    }
});

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 9;
gridBagConstraints.gridy = 4;
gridBagConstraints.gridwidth = 5;
gridBagConstraints.gridheight = 5;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
gridBagConstraints.insets = new java.awt.Insets(40, 10, 38, 0);
getContentPane().add(cerrar, gridBagConstraints);

```

```

jScrollPane1.setHorizontalScrollBarPolicy(javax.swing.ScrollPaneConstants.HORIZONTAL_SCROLLBAR_ALWAYS);

```

```

jScrollPane1.setVerticalScrollBarPolicy(javax.swing.ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);

```

```

jScrollPane1.setFocusCycleRoot(true);
jScrollPane1.setMaximumSize(new java.awt.Dimension(4450, 2200));
jScrollPane1.setMinimumSize(new java.awt.Dimension(250, 200));
jScrollPane1.setPreferredSize(new java.awt.Dimension(250, 200));
jList1.setModel(lsexec);
jScrollPane1.setViewportViewView(jList1);

```

```

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 1;
gridBagConstraints.gridy = 0;
gridBagConstraints.gridwidth = 8;
gridBagConstraints.gridheight = 9;
gridBagConstraints.fill = java.awt.GridBagConstraints.BOTH;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
gridBagConstraints.weightx = 1.0;
gridBagConstraints.weighty = 1.0;
gridBagConstraints.insets = new java.awt.Insets(1, 1, 0, 0);
getContentPane().add(jScrollPane1, gridBagConstraints);

```

```

Excepcion.setColumns(20);
Excepcion.setMinimumSize(new java.awt.Dimension(45, 22));
Excepcion.setPreferredSize(new java.awt.Dimension(226, 30));
gridBagConstraints = new java.awt.GridBagConstraints();

```

```

gridBagConstraints.gridx = 1;
gridBagConstraints.gridy = 10;
gridBagConstraints.ipadx = 59;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
gridBagConstraints.insets = new java.awt.Insets(20, 34, 0, 20);
getContentPane().add(Excepcion, gridBagConstraints);

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 0;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
getContentPane().add(jLabell, gridBagConstraints);

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 1;
gridBagConstraints.gridy = 9;
getContentPane().add(error, gridBagConstraints);

java.awt.Dimension screenSize =
java.awt.Toolkit.getDefaultToolkit().getScreenSize();
setBounds((screenSize.width-413)/2, (screenSize.height-327)/2, 413, 327);
}
// </editor-fold>

private void eliminarMouseClicked(java.awt.event.MouseEvent evt) {
    int indic=jList1.getSelectedIndex(); // Posicion de JList seleccionado
    if (indic!=-1){ // se ha seleccionado un elemento

jlistexc.quitarSet(jlistexc.ListaExcepciones,jList1.getSelectedValue().toString());
//Quitar del conjunto Excepciones
        lsexc.removeElementAt(indic); //Quitar de JList
    }
}

private void añadirMouseClicked(java.awt.event.MouseEvent evt) {
    error.setText(" "); //Borra el indicador del error
    String elemento=Excepcion.getText(); //toma la palabra tecleada
    String inicial=elemento.substring(0,1);
    String cadenaMAY=elemento.toLowerCase();
    String INICMAY=cadenaMAY.substring(0,1);
    if (inicial.equals(INICMAY)==false) { //Comprueba que empieza por mayusculas
        jlistexc.añadirSet(jlistexc.ListaExcepciones,elemento); //a ade la
palabra al conjunto
        lsexc.clear(); //borra la pantalla
        creaJLIST(); // se vuelve a crear y mostrar la JList para que salga
ordenada
    } else {
        error.setText("No es un Nombre Propio. No empieza por mayusculas");
    }
}

private void cerrarMouseClicked(java.awt.event.MouseEvent evt) {
    try { //Guarda el contenido de la JList en el fichero Excepciones.txt
        BufferedWriter out= new BufferedWriter(new
FileWriter("Excepciones.txt"));
        out.write(jlistexc.ListaExcepciones.toString());
        out.close();
    } catch (IOException e){
    }
    lsexc.clear(); //Limpia el contenido de JList
    this.dispose(); //Cierra y

```

```

        this.setVisible(false); //oculta la ventana
    }

    public static void creaJLIST() { //Pone los elemetos del conjunto
Excepciones en el Jlist
        Iterator c=jlistexc.ListaExcepciones.iterator();
        while (c.hasNext()==true) {
            lsexc.addElement(c.next());
        }
    }
    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                creaJLIST();
                new EditarExcepciones().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JTextField Excepcion;
    private javax.swing.JButton añadir;
    private javax.swing.JButton cerrar;
    private javax.swing.JButton eliminar;
    private javax.swing.JLabel error;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JList jList1;
    private javax.swing.JScrollPane jScrollPane1;
    // End of variables declaration
}

```

```

/*
 * EditarNPConocidos.java
 *
 * Created on 9 de abril de 2006, 20:23
 */

package nombrespropios;
import javax.swing.*;
import java.util.*;
import java.io.*;
/**
 *
 * @author Victor del Canto
 */
public class EditarNPConocidos extends javax.swing.JFrame {
    public static DefaultListModel lscono=new DefaultListModel();
    public static EditorNP jlistcon= new EditorNP();
    /** Creates new form EditarNPConocidos */
    public EditarNPConocidos() {
        initComponents();
    }

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

        añadir = new javax.swing.JButton();
        eliminar = new javax.swing.JButton();
        cerrar = new javax.swing.JButton();
        jScrollPane1 = new javax.swing.JScrollPane();
        jList1 = new javax.swing.JList();
        Nombre = new javax.swing.JTextField();
        error = new javax.swing.JLabel();

        getContentPane().setLayout(new java.awt.GridBagLayout());

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("EDITAR NP CONOCIDOS");
        setFocusCycleRoot(false);
        añadir.setText("A\u00f1adir");
        añadir.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                añadirMouseClicked(evt);
            }
        });

        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 6;
        gridBagConstraints.gridy = 14;
        gridBagConstraints.insets = new java.awt.Insets(10, 10, 10, 10);
        getContentPane().add(añadir, gridBagConstraints);

        eliminar.setText("Eliminar");
        eliminar.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                eliminarMouseClicked(evt);
            }
        });
    }

```



```

    }
});

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 10;
gridBagConstraints.gridy = 14;
gridBagConstraints.insets = new java.awt.Insets(10, 10, 10, 10);
getContentPane().add(eliminar, gridBagConstraints);

cerrar.setText("Cerrar");
cerrar.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        cerrarMouseClicked(evt);
    }
});

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 8;
gridBagConstraints.gridy = 14;
gridBagConstraints.insets = new java.awt.Insets(10, 10, 10, 10);
getContentPane().add(cerrar, gridBagConstraints);

jScrollPane1.setHorizontalScrollBarPolicy(javax.swing.ScrollPaneConstants.HORIZONTAL_SCROLLBAR_ALWAYS);

jScrollPane1.setVerticalScrollBarPolicy(javax.swing.ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);
jScrollPane1.setInheritsPopupMenu(true);
jScrollPane1.setMaximumSize(new java.awt.Dimension(4450, 2200));
jScrollPane1.setMinimumSize(new java.awt.Dimension(450, 200));
jScrollPane1.setPreferredSize(new java.awt.Dimension(250, 200));
jList1.setModel(lscono);
jScrollPane1.setViewportViewView(jList1);

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 1;
gridBagConstraints.gridy = 2;
gridBagConstraints.gridwidth = 11;
gridBagConstraints.gridheight = 11;
getContentPane().add(jScrollPane1, gridBagConstraints);

Nombre.setColumns(20);
Nombre.setMinimumSize(new java.awt.Dimension(60, 22));
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 1;
gridBagConstraints.gridy = 14;
gridBagConstraints.gridwidth = 3;
gridBagConstraints.ipadx = 59;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTHWEST;
gridBagConstraints.insets = new java.awt.Insets(10, 34, 10, 10);
getContentPane().add(Nombre, gridBagConstraints);

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 4;
gridBagConstraints.gridy = 17;
getContentPane().add(error, gridBagConstraints);

java.awt.Dimension screenSize =
java.awt.Toolkit.getDefaultToolkit().getScreenSize();
setBounds((screenSize.width-450)/2, (screenSize.height-300)/2, 450, 300);
}

```

```
// </editor-fold>

private void eliminarMouseClicked(java.awt.event.MouseEvent evt) {
    int indic=jList1.getSelectedIndex(); //Posicion de JList seleccionado
    if (indic!=-1){ //se ha seleccionado un elemento

jlistcon.quitarSet(jlistcon.ListaConocidos,jList1.getSelectedValue().toString());
//quitar del conjunto Conocidos
        lscono.removeElementAt(indic); //quitar de JList
    }
}

private void añadirMouseClicked(java.awt.event.MouseEvent evt) {
    error.setText(" "); //borra el indicador de error
    String elemento=Nombre.getText(); //toma la palabra tecleada
    String inicial=elemento.substring(0,1);
    String cadenaMAY=elemento.toLowerCase();
    String INICMAY=cadenaMAY.substring(0,1);
    if (inicial.equals(INICMAY)==false) { //Comprueba que empieza por
mayusculas
        jlistcon.añadirSet(jlistcon.ListaConocidos,elemento); //a ade la
palabra al conjunto Conocidos
        lscono.clear(); //borra la pantalla
        creaJLIST(); // se vuelve a crear y mostrar la JList para que
salga ordenada
    } else {
        error.setText("No es un Nombre Propio. No empieza por mayusculas");
    }
}

private void cerrarMouseClicked(java.awt.event.MouseEvent evt) {
    try { //guarda el contenido de la JList en el fichero Conocidos.txt
        BufferedWriter out= new BufferedWriter(new FileWriter("Conocidos.txt"));
        out.write(jlistcon.ListaConocidos.toString());
        out.close();
    } catch (IOException e){
    }
    System.out.println(jlistcon.ListaConocidos.toString());
    lscono.clear(); //limpia el contenido de la JList
    this.dispose(); //cierra y
    this.setVisible(false); //oculta la ventana
}

public static void creaJLIST() { //Pone los elementos del conjunto
Conocidos en un JList
    Iterator c=jlistcon.ListaConocidos.iterator();
    while (c.hasNext()==true) {
        lscono.addElement(c.next());
    }
}
/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            creaJLIST();
            new EditarNPConocidos().setVisible(true);
        }
    });
}

```

```
// Variables declaration - do not modify
private javax.swing.JTextField Nombre;
private javax.swing.JButton añadir;
private javax.swing.JButton cerrar;
private javax.swing.JButton eliminar;
private javax.swing.JLabel error;
private javax.swing.JList jList1;
private javax.swing.JScrollPane jScrollPane1;
// End of variables declaration

}
```

```

/*
 * ExplorarArchivo.java
 *
 * Created on 6 de abril de 2006, 17:10
 */

package nombresproprios;

/**
 *
 * @author Victor del Canto
 */
public class ExplorarArchivo extends javax.swing.JFrame {

    /** Creates new form ExplorarArchivo */
    public ExplorarArchivo() {
        initComponents();
    }

    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    // <editor-fold defaultstate="collapsed" desc=" Generated Code ">
    private void initComponents() {
        jFileChooser1 = new javax.swing.JFileChooser();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("Abrir texto");
        getContentPane().add(jFileChooser1, java.awt.BorderLayout.CENTER);

        java.awt.Dimension screenSize =
java.awt.Toolkit.getDefaultToolkit().getScreenSize();
        setBounds((screenSize.width-400)/2, (screenSize.height-300)/2, 400, 300);
    }
    // </editor-fold>

    /**
     * @param args the command line arguments
     */
    public String fichero(ExplorarArchivo explordialog) {
        int returnVal=jFileChooser1.showOpenDialog(explordialog); //valor
asignado de abrir o cancelar
        if (returnVal==jFileChooser1.APPROVE_OPTION) { //si se ha
pulsado abrir
            String nombre=new String(jFileChooser1.getSelectedFile().getPath());
//devuelve el nombre de archivo a
            return(nombre); //tratar con su trayectoria
        } else {
            return("sin fichero"); //si se ha pulsado cancelar
        }
    }

    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new ExplorarArchivo().setVisible(true);
            }
        });
    }
}

```

```
// Variables declaration - do not modify
private javax.swing.JFileChooser jFileChooser1;
// End of variables declaration

}
```

```

/*
 * ListarNPconocidos.java
 *
 * Created on 10 de abril de 2006, 10:37
 */

package nombrespropios;
import javax.swing.*;
import java.util.*;

/**
 *
 * @author Victor del Canto
 */
public class ListarNPconocidos extends javax.swing.JFrame {

    public static DefaultListModel lscon=new DefaultListModel();
    public static EditorNP jlistcon= new EditorNP();

    /** Creates new form ListarNPconocidos */
    public ListarNPconocidos() {
        initComponents();
    }

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

        jScrollPane1 = new javax.swing.JScrollPane();
        jList1 = new javax.swing.JList();
        Cerrar = new javax.swing.JButton();

        getContentPane().setLayout(new java.awt.GridBagLayout());

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("LISTA DE NOMBRES PROPIOS CONOCIDOS");

        jScrollPane1.setHorizontalScrollBarPolicy(javax.swing.ScrollPaneConstants.HORIZONTAL_SCROLLBAR_ALWAYS);

        jScrollPane1.setVerticalScrollBarPolicy(javax.swing.ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);
        jScrollPane1.setMaximumSize(new java.awt.Dimension(4450, 2200));
        jScrollPane1.setMinimumSize(new java.awt.Dimension(250, 200));
        jScrollPane1.setPreferredSize(new java.awt.Dimension(375, 280));
        jList1.setModel(lscon);
        jList1.setSelectionMode(javax.swing.ListSelectionModel.SINGLE_SELECTION);
        jScrollPane1.setViewportView(jList1);

        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 1;
        gridBagConstraints.gridy = 3;
        gridBagConstraints.gridwidth = 35;

```



```

        gridBagConstraints.gridheight = 12;
        gridBagConstraints.ipadx = 60;
        gridBagConstraints.ipady = 60;
        getContentPane().add(jScrollPane, gridBagConstraints);

        Cerrar.setText("Cerrar");
        Cerrar.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                CerrarMouseClicked(evt);
            }
        });

        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 14;
        gridBagConstraints.gridy = 17;
        gridBagConstraints.gridwidth = 8;
        gridBagConstraints.insets = new java.awt.Insets(10, 100, 1, 4);
        getContentPane().add(Cerrar, gridBagConstraints);

        java.awt.Dimension screenSize =
        java.awt.Toolkit.getDefaultToolkit().getScreenSize();
        setBounds((screenSize.width-450)/2, (screenSize.height-350)/2, 450, 350);
    }
    // </editor-fold>

    private void CerrarMouseClicked(java.awt.event.MouseEvent evt) {
        lscon.clear(); //borra el contenido de la JList
        this.dispose(); //cierra y
        this.setVisible(false); //oculta la ventana
    }

    public static void creaJLIST() {
        //Pone los elementos del conjunto Conocidos que estan en el texto en un JList
        Iterator c=jlistcon.ListaConocidos.iterator();
        while (c.hasNext()==true) {
            Object palabra=c.next();
            if (jlistcon.lista.contains(palabra)==true) {
                lscon.addElement(palabra);
            }
        }
    }
    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                creaJLIST();
                new ListarNPconocidos().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JButton Cerrar;
    public static javax.swing.JList jList1;
    private javax.swing.JScrollPane jScrollPane1;
    // End of variables declaration
}

```

C:\WINDOWS\NombresPropios\src\nombrespropios\ListarNPdesconocidos.java

```

/*
 * ListarNPdesconocidos.java
 *
 * Created on 10 de abril de 2006, 10:49
 */

package nombrespropios;
import javax.swing.*;
import java.util.*;
import java.io.*;

/**
 *
 * @author Victor del Canto
 */
public class ListarNPdesconocidos extends javax.swing.JFrame {

    public static DefaultListModel lsd= new DefaultListModel();
    public static EditorNP jlistdesc= new EditorNP();
    /** Creates new form ListarNPdesconocidos */
    public ListarNPdesconocidos() {
        initComponents();
    }

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

        jScrollPane1 = new javax.swing.JScrollPane();
        jList1 = new javax.swing.JList();
        Cerrar = new javax.swing.JButton();
        Añadir = new javax.swing.JButton();
        AñadirExcep = new javax.swing.JButton();
        Error = new javax.swing.JLabel();

        getContentPane().setLayout(new java.awt.GridBagLayout());

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("LISTA DE NOMBRES PROPIOS DESCONOCIDOS");

        jScrollPane1.setHorizontalScrollBarPolicy(javax.swing.ScrollPaneConstants.HORIZONTAL_SCROLLBAR_ALWAYS);

        jScrollPane1.setVerticalScrollBarPolicy(javax.swing.ScrollPaneConstants.VERTICAL_SCROLLBAR_ALWAYS);
        jScrollPane1.setMaximumSize(new java.awt.Dimension(4450, 2200));
        jScrollPane1.setMinimumSize(new java.awt.Dimension(450, 200));
        jScrollPane1.setPreferredSize(new java.awt.Dimension(275, 200));
        jList1.setModel(lsd);
        jScrollPane1.setViewportView(jList1);

        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 1;
        gridBagConstraints.gridy = 1;
        gridBagConstraints.gridwidth = 25;
        gridBagConstraints.gridheight = 12;

```

```

gridBagConstraints.ipadx = 60;
gridBagConstraints.ipady = 60;
getContentPane().add(jScrollPane, gridBagConstraints);

Cerrar.setText("Cerrar");
Cerrar.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        CerrarMouseClicked(evt);
    }
});

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 11;
gridBagConstraints.gridy = 15;
gridBagConstraints.gridwidth = 8;
gridBagConstraints.insets = new java.awt.Insets(10, 2, 3, 6);
getContentPane().add(Cerrar, gridBagConstraints);

Añadir.setText("A\u00f1adir a NP Conocidos");
Añadir.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        AñadirMouseClicked(evt);
    }
});

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 4;
gridBagConstraints.gridy = 15;
gridBagConstraints.insets = new java.awt.Insets(10, 20, 2, 2);
getContentPane().add(Añadir, gridBagConstraints);

AñadirExcep.setText("A\u00f1adir a Excepciones");
AñadirExcep.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        AñadirExcepMouseClicked(evt);
    }
});

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 7;
gridBagConstraints.gridy = 15;
gridBagConstraints.insets = new java.awt.Insets(10, 2, 2, 2);
getContentPane().add(AñadirExcep, gridBagConstraints);

gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 4;
gridBagConstraints.gridy = 17;
gridBagConstraints.gridwidth = 10;
getContentPane().add(Error, gridBagConstraints);

java.awt.Dimension screenSize =
java.awt.Toolkit.getDefaultToolkit().getScreenSize();
setBounds((screenSize.width-450)/2, (screenSize.height-350)/2, 450, 350);
}
// </editor-fold>

private void AñadirMouseClicked(java.awt.event.MouseEvent evt) {
    int indic=jList1.getSelectedIndex(); //Existe un elemento
    if (indic!=-1){ // de JList seleccionado
        //Se a\u00f1ade al conjunto Conocidos

jlistdesc.a\u00f1adirSet(jlistdesc.ListaConocidos,jList1.getSelectedValue().toString());

```

```

        //Se quita el elemento del conjunto Desconocidos
jlistdesc.quitarSet(jlistdesc.ListaDesconocidos,jList1.getSelectedValue() .toString());
        //Se borra el elemento de la JList
        lsdes.removeElementAt(indic);
    }
}

private void AñadirExcepMouseClicked(java.awt.event.MouseEvent evt) {
    int indic=jList1.getSelectedIndex();    //Existe un elemento
    if (indic!=-1){                          //de la JList seleccionado
        // Se añade al conjunto Excepciones

jlistdesc.añadirSet(jlistdesc.ListaExcepciones,jList1.getSelectedValue() .toString());
        // Se quita el elemento del conjunto Desconocidos

jlistdesc.quitarSet(jlistdesc.ListaDesconocidos,jList1.getSelectedValue() .toString());
        // Se borra el elemento de la JList
        lsdes.removeElementAt(indic);
    }
}

private void CerrarMouseClicked(java.awt.event.MouseEvent evt) {
//Por si se han añadido elementos del conjunto Desconocidos a Conocidos o
Excepciones
    try {
        BufferedWriter outexc= new BufferedWriter(new
FileWriter("Excepciones.txt"));
        outexc.write(jlistdesc.ListaExcepciones.toString());    //guarda los
        elementos del conjunto Excepciones
        outexc.close();    //en el fichero Excepciones.txt
        BufferedWriter outcon= new BufferedWriter(new
FileWriter("Conocidos.txt"));
        outcon.write(jlistdesc.ListaConocidos.toString());    //guarda los
        elementos dl conjunto Conocidos
        outcon.close();    //en el fichero Conocidos.txt
    }catch (IOException e){
    }
    lsdes.clear();    //borra el contenido de la JList
    this.dispose();    //cierra y
    this.setVisible(false);    //oculta la ventana
}

public static void creaJLIST() {
//Pone los elementos que estan en el texto y no son del conjunto Conocidos ni
Excepciones en un JList
    Iterator c=jlistdesc.ListaDesconocidos.iterator();
    while (c.hasNext()==true) {
        Object palabra=c.next();
        if (jlistdesc.lista.contains(palabra)==true) {
            lsdes.addElement(palabra);
        }
    }
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {

```

```

        creaJLIST ();
        new ListarNPdesconocidos ().setVisible (true) ;

    }
}

// Variables declaration - do not modify
private javax.swing.JButton Añadir;
private javax.swing.JButton AñadirExcep;
private javax.swing.JButton Cerrar;
private javax.swing.JLabel Error;
public static javax.swing.JList jList1;
private javax.swing.JScrollPane jScrollPane1;
// End of variables declaration

}

```

C:\WINDOWS\NombresPropios\src\nombrespropios>ErrorFichero.java

/*

```
* ErrorFichero.java
*
* Created on 5 de abril de 2006, 21:29
*/
```

```
package nombresproprios;
```

```
/**
```

```
*
```

```
* @author Victor del Canto
```

```
*/
```

```
public class ErrorFichero extends javax.swing.JDialog {
```

```
    /** Creates new form ErrorFichero */
```

```
    public ErrorFichero(java.awt.Frame parent, boolean modal) {
        super(parent, modal);
        initComponents();
    }
```

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

```
    // <editor-fold defaultstate="collapsed" desc=" Generated Code ">
```

```
    private void initComponents() {
        java.awt.GridBagConstraints gridBagConstraints;
```

```
        jLabel1 = new javax.swing.JLabel();
        cerrar = new javax.swing.JButton();
        jLabel2 = new javax.swing.JLabel();
```

```
        getContentPane().setLayout(new java.awt.GridBagLayout());
```

```
        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("ERROR DE ARCHIVO");
        setBackground(new java.awt.Color(255, 0, 0));
        setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
        setResizable(false);
```

```
        jLabel1.setText("ERROR DE I/O. No se ha encontrado o no se ha podido
abrir el fichero");
```

```
        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 3;
        gridBagConstraints.gridy = 3;
        gridBagConstraints.gridwidth = 10;
        getContentPane().add(jLabel1, gridBagConstraints);
```

```
        cerrar.setText("CERRAR");
        cerrar.setHorizontalAlignment(javax.swing.SwingConstants.TRAILING);
        cerrar.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
        cerrar.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                cerrarMouseClicked(evt);
            }
        });
```

```
        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 10;
        gridBagConstraints.gridy = 10;
        gridBagConstraints.insets = new java.awt.Insets(2, 225, 2, 2);
        getContentPane().add(cerrar, gridBagConstraints);
```



```

        jLabel2.setText ("
");

        gridBagConstraints = new java.awt.GridBagConstraints ();
        gridBagConstraints.gridx = 3;
        gridBagConstraints.gridy = 4;
        gridBagConstraints.gridwidth = 10;
        gridBagConstraints.gridheight = 5;
        getContentPane().add(jLabel2, gridBagConstraints);

        java.awt.Dimension screenSize =
java.awt.Toolkit.getDefaultToolkit().getScreenSize();
        setBounds((screenSize.width-400)/2, (screenSize.height-300)/2, 400, 300);
    }
    // </editor-fold>

    private void cerrarMouseClicked(java.awt.event.MouseEvent evt) {
        this.dispose(); //cierra y
        this.setVisible(false); //oculta la ventana
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new ErrorFichero(new javax.swing.JFrame(), true).setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton cerrar;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    // End of variables declaration
}

```