



Universidad de Córdoba



ESCUELA POLITÉCNICA SUPERIOR DE CÓRDOBA

INGENIERÍA INFORMÁTICA

---

Proyecto de fin de carrera

**SimAS: Simulador de Análisis Sintácticos**

---

MANUAL DE CÓDIGO

***Autora***

Vanesa González Pérez

***Director***

Dr. Nicolás Luis Fernández García

Córdoba, Junio de 2015



# Índice general

<b>1. Documentación externa</b>	<b>1</b>
1.1. Introducción . . . . .	1
1.2. Recursos . . . . .	1
1.2.1. Recursos de Hardware . . . . .	2
1.2.2. Recursos de Software . . . . .	2
1.3. Entorno de Desarrollo . . . . .	3
1.3.1. Preparación del entorno de desarrollo y de la Máquina Virtual de Java . . . . .	3
1.3.2. Compilación de la aplicación . . . . .	4
1.3.3. Empaquetado de la aplicación . . . . .	6
1.4. Descripción modular . . . . .	7
 <b>2. Documentacion interna</b>	 <b>11</b>
2.1. Introducción . . . . .	11
2.2. Paquete SimAS . . . . .	11
2.2.1. Bienvenida.java . . . . .	11
2.2.2. SimAS.java . . . . .	14

2.3. Paquete Centro de ayuda . . . . .	16
2.3.1. AcercaDe.java . . . . .	16
2.4. Paquete Editor . . . . .	20
2.4.1. ColCanLALR.java . . . . .	20
2.4.2. ColCanLR0.java . . . . .	24
2.4.3. ColCanLR1.java . . . . .	30
2.4.4. ConjElementosLALR.java . . . . .	35
2.4.5. ConjElementosLR0.java . . . . .	43
2.4.6. ConjElementosLR1.java . . . . .	48
2.4.7. Editor.java . . . . .	56
2.4.8. ElementosLALR.java . . . . .	75
2.4.9. ElementosLR0.java . . . . .	76
2.4.10. ElementosLR1.java . . . . .	77
2.4.11. FuncionError.java . . . . .	78
2.4.12. PanelCreacionGramaticaPaso1.java . . . . .	79
2.4.13. PanelCreacionGramaticaPaso2.java . . . . .	85
2.4.14. PanelCreacionGramaticaPaso3.java . . . . .	91
2.4.15. PanelCreacionGramaticaPaso4.java . . . . .	97
2.4.16. ParteAccion.java . . . . .	102
2.4.17. ParteIrA.java . . . . .	115
2.4.18. TablaLR.java . . . . .	120
2.4.19. TablaPredictiva.java . . . . .	121

---

2.4.20. VentanaCreacionGramatica.java . . . . .	125
2.4.21. VentanaProducciones.java . . . . .	128
2.4.22. VentanaResultadosValidacion.java . . . . .	144
2.4.23. VentanaSimbolosNoTerminales.java . . . . .	146
2.4.24. VentanaSimbolosTerminales.java . . . . .	154
2.4.25. Visualizacion.java . . . . .	172
2.5. Paquete Simulador . . . . .	178
2.5.1. CadEntrada.java . . . . .	178
2.5.2. MiRender.java . . . . .	184
2.5.3. NuevaFuncionError.java . . . . .	185
2.5.4. NuevaSimulacionAsc.java . . . . .	193
2.5.5. NuevaSimulacionDesc.java . . . . .	208
2.5.6. PanelNuevaSimAscPaso1.java . . . . .	222
2.5.7. PanelNuevaSimAscPaso2.java . . . . .	226
2.5.8. PanelNuevaSimAscPaso3.java . . . . .	232
2.5.9. PanelNuevaSimAscPaso4.java . . . . .	237
2.5.10. PanelNuevaSimAscPaso5.java . . . . .	243
2.5.11. PanelNuevaSimAscPaso6.java . . . . .	252
2.5.12. PanelNuevaSimDescPaso1.java . . . . .	261
2.5.13. PanelNuevaSimDescPaso2.java . . . . .	272
2.5.14. PanelNuevaSimDescPaso3.java . . . . .	279
2.5.15. PanelNuevaSimDescPaso4.java . . . . .	284

2.5.16. PanelNuevaSimDescPaso5.java . . . . .	292
2.5.17. Simulador.java . . . . .	301
2.5.18. VentanaSimuladorAsc.java . . . . .	325
2.5.19. VentanaSimuladorDesc.java . . . . .	329
2.6. Paquete Gramática . . . . .	333
2.6.1. Antecedente.java . . . . .	333
2.6.2. Consecuente.java . . . . .	334
2.6.3. Gramatica.java . . . . .	335
2.6.4. NoTerminal.java . . . . .	367
2.6.5. Produccion.java . . . . .	368
2.6.6. Simbolo.java . . . . .	370
2.6.7. Terminal.java . . . . .	372

## Referencias

**375**

# Índice de figuras

1.1. Añadir las librerías iText . . . . .	4
1.2. Compilación de SimAS . . . . .	5
1.3. Ejecución de SimAS . . . . .	5
1.4. Empaquetado de SimAS . . . . .	6





# Capítulo 1

## Documentación externa

### 1.1. Introducción

Este manual de código corresponde a la aplicación **SimAS**, *Simulador de análisis Sintácticos*. SimAS es una aplicación didáctica y multiplataforma que permite crear y editar gramáticas de contexto libre, con las que se podrá realizar la simulación del análisis sintáctico descendente y del análisis sintáctico ascendente. En este manual se describen los ficheros de código fuente necesarios para la construcción de la aplicación.

En este manual se detallan los recursos de software y hardware necesarios para la implementación del sistema, además se presentan todos los archivos que forman parte de la aplicación, organizados según su funcionalidad y describiendo su contenido.

### 1.2. Recursos

Para el desarrollo de la aplicación se han utilizado los siguientes recursos:

- Recursos de Hardware.
- Recursos de Software.

### 1.2.1. Recursos de Hardware

Los recursos hardware se pueden subdividir en dos categorías:

- Recursos de hardware para el desarrollo del proyecto: utilizados en el desarrollo de la aplicación.
  - Ordenador portátil Asus PRO5IJSeries. Intel Core i3-370M, 2,4 GHz. 4 GB de memoria RAM. HDD de 500 GB. Tarjeta gráfica Radeon HD 6370m.
- Recursos hardware para el uso de la aplicación: requisitos mínimos recomendados para el buen funcionamiento de la aplicación.
  - Requisitos mínimos declarados por Sun Microsystems para que la Máquina Virtual de Java pueda funcionar correctamente.

### 1.2.2. Recursos de Software

- Sistemas operativos:
  - Ubuntu Linux, versión 12.10
  - Microsoft Windows 7
- Este programa está desarrollado en lenguaje **Java** (Java SE (JDK) 7u40).
- Editores: editor de textos Texmaker para Latex.
- Entornos de desarrollo. NetBeans 7.3.1. Se utilizará para la codificación y pruebas de la aplicación.
- Dentro de NetBeans se ha utilizado *Java Swing* para el desarrollo de la interfaz de usuario de SimAS.
- La aplicación SimAS permite la creación de informes de la gramática de contexto libre y de la simulación de ésta. Estos informes se crean mediante la librería *iText* 5.5.0 de Java.
- SimAS también permite guardar y cargar gramáticas en formato *XML*.

### 1.3. Entorno de Desarrollo

Como se ha comentado anteriormente, el entorno de desarrollo usado es NetBeans IDE 7.3.1. Con esta herramienta se ha desarrollado el código fuente de la aplicación, se han hecho las pruebas, y se ha generado el paquete final ejecutable. Debido a que el lenguaje usado para el desarrollo ha sido Java, esta aplicación tiene la característica de ser multiplataforma y multiarquitectura, es decir, que se puede ejecutar de manera independiente, y sin hacer ninguna modificación, en sistemas con arquitectura distinta y con sistemas operativos diferentes, obteniendo el mismo resultado.

Esto es posible debido a que el intérprete del lenguaje Java, *Java Virtual Machine* (JVM), está instalado en la máquina que se ejecuta; si esto no fuera así, no podría ejecutarse el programa. Así pues, instalando en cada sistema operativo una versión compatible del intérprete, y teniendo en cuenta también la arquitectura del ordenador, se puede ejecutar un mismo programa en cualquier sistema que cuente con este intérprete.

Para llevar a cabo el desarrollo de esta aplicación se han seguido los siguientes pasos:

1. Preparación del entorno de desarrollo y de la Máquina Virtual de Java.
2. Compilación de la aplicación.
3. Empaquetado de la aplicación.

#### 1.3.1. Preparación del entorno de desarrollo y de la Máquina Virtual de Java

Los pasos seguidos para preparar y configurar el entorno de desarrollo y la máquina virtual son los siguientes:

1. Descargar Netbeans con Java SE (actualmente, a la hora de escribir este manual, se encuentra disponible la versión 8.0.2) de la página web [1].
2. Instalar el archivo descargado (nótese que NetBeans es portable y la propia página web detecta el sistema operativo de la máquina, para bajarse un instalador de NetBeans apropiado, para Windows, Linux o Mac).

3. Crear un proyecto en NetBeans y añadir el código fuente de SimAS a la carpeta src.
4. Añadir el archivo .jar de la librería iText (iText-5.5.0.jar) a las referencias del proyecto. Para esto, en la jerarquía de directorios, del proyecto, se hace clic derecho sobre la opción Libraries y después se selecciona la opción Add JAR/Folder para seleccionar la librería a añadir. Se recomienda tener bien localizada esta librería e incluirla en la carpeta de desarrollo del proyecto (para evitar que falle el proyecto si la librería es borrada por error).

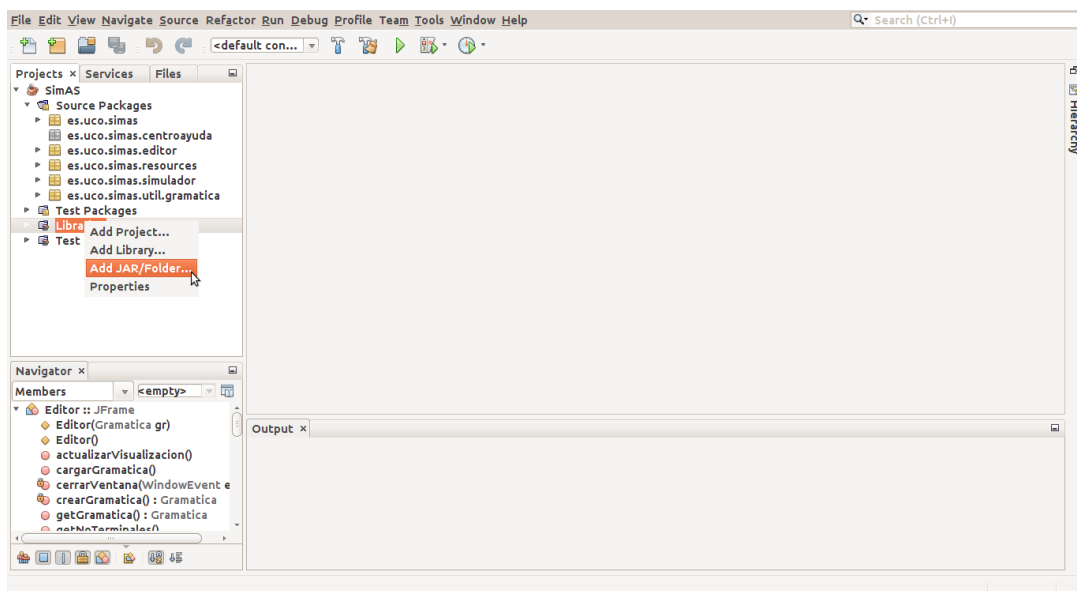


Figura 1.1: Añadir las librerías iText

### 1.3.2. Compilación de la aplicación

Para compilar el programa y ejecutar la aplicación en NetBeans, sólo hay que pulsar el botón Run tal y como se muestra en la figura 1.2. NetBeans cargará y ejecutará la clase principal de la aplicación directamente (compilándose antes todos los ficheros .java a .class).

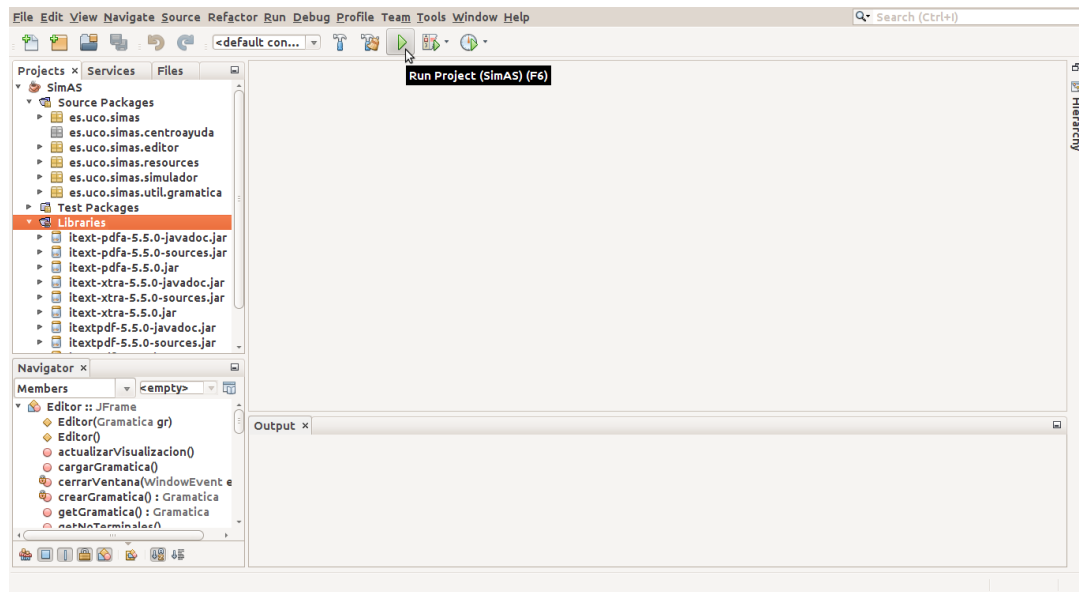


Figura 1.2: Compilación de SimAS

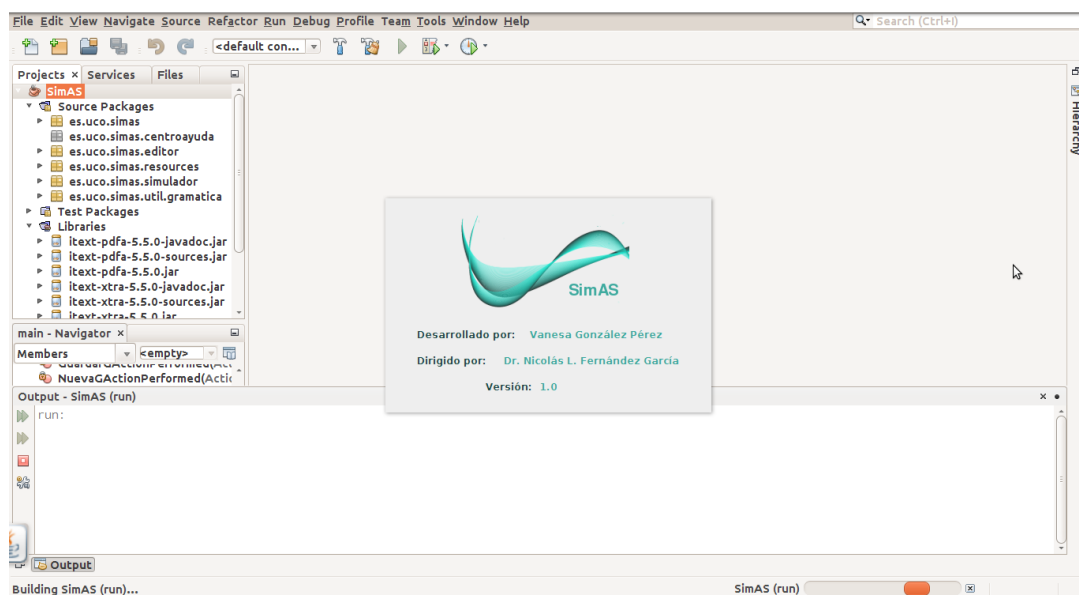


Figura 1.3: Ejecución de SimAS

### 1.3.3. Empaquetado de la aplicación

El último paso para obtener la aplicación ejecutable, es empaquetar el código para generar el archivo en formato .jar. Para ello, se ha de usar la opción *Clean and Build Project* del menú Run de Netbeans.

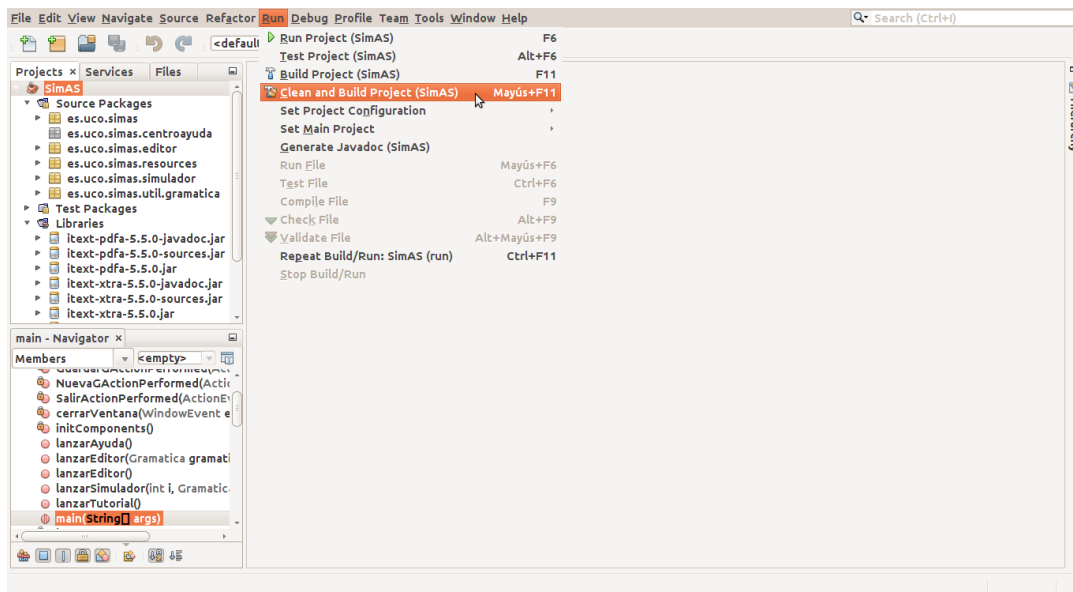


Figura 1.4: Empaquetado de SimAS

Una vez hecho esto, en la carpeta del proyecto aparecerá una carpeta denominada *dist*, que contendrá el archivo autoejecutable del proyecto, **SimAS.jar**, así como la carpeta *lib*, que tendrá las librerías utilizadas por el proyecto:

- **AbsoluteLayout**: utilizada por las interfaces gráficas (la añade NetBeans automáticamente).
- **BeansBinding**: utilizada por NetBeans para la construcción del ejecutable (la añade NetBeans automáticamente).
- **iTextPdf**: librería para la generación de documentos PDF (fue añadida al proyecto tal y como se explicó anteriormente).

Existen dos formas de ejecutar la aplicación fuera de NetBeans:

1. Copiar la carpeta *dist* y ejecutar **SimAS.jar** haciendo doble clic con el botón derecho y luego ejecutar la máquina virtual.

2. Abrir un terminal y utilizar la instrucción `java -jar SimAS.jar`.

Es importante tener en cuenta que la carpeta `lib` siempre deberá ir junto con el ejecutable de la aplicación, puesto que si se borra, se renombra o se elimina algún componente de la carpeta, la ejecución del programa fallará al no encontrar las librerías necesarias.

## 1.4. Descripción modular

El proyecto está dividido en paquetes, los cuales contienen los ficheros con el código fuente y los recursos de la aplicación. Los paquetes que componen el programa son:

- **Paquete `es.uco.simas`:** es el paquete principal de la aplicación, y el que contiene los demás paquetes y la main-class de la aplicación.
  - *SimAS.java*: objeto principal de la aplicación. Es la encargada de lanzar cada uno de los módulos: Editor, Simulador y Ayuda.
  - *Bienvenida.java*: es el panel de bienvenida a la aplicación, donde se muestra el logotipo y la versión de la misma.
- **Paquete `es.uco.simas.centroayuda`:**
  - Recursos de la ayuda: ficheros en formato `.html` e imágenes. Es el contenido de la ayuda.
  - *AcercaDe.java*: información sobre la aplicación.
- **Paquete `es.uco.simas.editor`:**
  - *ColCanLALR.java*: calcula la colección canónica de elementos LALR.
  - *ColCanLR0.java*: calcula la colección canónica de elementos LR(0).
  - *ColCanLR1.java*: calcula la colección canónica de elementos LR(1).
  - *ConjElementosLALR.java* calcula el conjunto de elementos LALR.
  - *ConjElementosLR0.java* calcula el conjunto de elementos LR(0).
  - *ConjElementosLR1.java* calcula el conjunto de elementos LR(1).
  - *Editor.java*: ventana del editor.
  - *ElementosLALR.java*: calcula los elementos LALR.
  - *ElementosLR0.java*: calcula los elementos LR(0).

- *ElementosLR1.java*: calcula los elementos LR(1).
  - *FuncionError.java*: crea las funciones de error.
  - *PanelCreacionGramaticaPaso1.java*: paso número uno del asistente de creación de gramáticas.
  - *PanelCreacionGramaticaPaso2.java*: paso número dos del asistente de creación de gramáticas.
  - *PanelCreacionGramaticaPaso3.java*: paso número tres del asistente de creación de gramáticas.
  - *PanelCreacionGramaticaPaso4.java*: paso número cuatro del asistente de creación de gramáticas.
  - *ParteAccion.java*: parte acción de la tabla LR.
  - *ParteIrA.java*: parte ir\_a de la tabla LR.
  - *TablaLR.java*: clase que almacena los datos de la tabla LR para el análisis sintáctico ascendente.
  - *TablaPredictiva.java*: clase que almacena los datos de la tabla predictiva para el análisis sintáctico descendente.
  - *VentanaCreacionGramatica.java*: ventana para la creación de la gramática de contexto libre.
  - *VentanaProducciones.java*: ventana para la definición de las producciones de la gramática de contexto libre.
  - *VentanaResultadosValidacion.java*: ventana que muestra los resultados de la validación de la gramática.
  - *VentanaSimbolosNoTerminales.java*: ventana para la definición de los símbolos no terminales de la gramática.
  - *VentanaSimbolosTerminales.java*: ventana para la definición de los símbolos terminales de la gramática.
  - *Visualizacion.java*: ventana para la visualización de la gramática.
- **Paquete es.uco.simas.resources:**
- Recursos de la interfaz: recursos gráficos de la interfaz (iconos, fuentes de letra, etcétera).
- **Paquete es.uco.simas.simulador:**
- *CadEntrada.java*: clase que almacena los datos de la cadena de entrada.
  - *MiRender.java*: clase para almacenar el formato para mostrar los datos en las tablas.
  - *NuevaFuncionError.java*: ventana para introducir los datos de una nueva función de error.



- *NuevaSimulacionAsc.java*: ventana para crear una nueva simulación ascendente.
- *NuevaSimulacionDesc.java*: ventana para crear una nueva simulación descendente.
- *PanelNuevaSimAscPaso1.java*: paso número uno del asistente de la simulación del método ascendente.
- *PanelNuevaSimAscPaso2.java*: paso número dos del asistente de la simulación del método ascendente.
- *PanelNuevaSimAscPaso3.java*: paso número tres del asistente de la simulación del método ascendente.
- *PanelNuevaSimAscPaso4.java*: paso número cuatro del asistente de la simulación del método ascendente.
- *PanelNuevaSimAscPaso5.java*: paso número cinco del asistente de la simulación del método ascendente.
- *PanelNuevaSimAscPaso6.java*: paso número seis del asistente de la simulación del método ascendente.
- *PanelNuevaSimDescPaso1.java*: paso número uno del asistente de la simulación del método descendente.
- *PanelNuevaSimDescPaso2.java*: paso número dos del asistente de la simulación del método descendente.
- *PanelNuevaSimDescPaso3.java*: paso número tres del asistente de la simulación del método descendente.
- *PanelNuevaSimDescPaso4.java*: paso número cuatro del asistente de la simulación del método descendente.
- *PanelNuevaSimDescPaso5.java*: paso número cinco del asistente de la simulación del método descendente.
- *Simulador.java*: clase que almacena los datos de la simulación.
- *VentanaSimuladorAsc.java*: ventana del simulador ascendente.
- *VentanaSimuladorDesc.java*: ventana del simulador descendente.

■ **Paquete `es.uco.simas.util.gramatica`:**

- *Antecedente.java*: clase que almacena los datos de los antecedentes de las producciones.
- *Consecuente.java*: clase que almacena los datos de los consecuentes de las producciones.
- *Gramatica.java*: clase que almacena los datos de la gramática.
- *NoTerminal.java*: clase que almacena los datos de los símbolos no terminales de la gramática.

- *Produccion.java*: clase que almacena los datos de las producciones de la gramática.
- *Simbolo.java*: clase que almacena los datos de los símbolos de la gramática.
- *Terminal.java*: clase que almacena los datos de los símbolos terminales de la gramática.

# Capítulo 2

## Documentacion interna

### 2.1. Introducción

En este capítulo se mostrará el código fuente de cada uno de los archivos creados para el desarrollo de la aplicación SimAS, dividido en función de los paquetes mostrados en el capítulo anterior.

### 2.2. Paquete SimAS

#### 2.2.1. Bienvenida.java

```
1 //SimAS
2 // Bienvenida
3
4 package es.uco.simas;
5
6 import java.util.logging.Level;
7 import java.util.logging.Logger;
8
9 /**
10  * @author vanesa
11  */
12 public class Bienvenida extends javax.swing.JFrame implements Runnable{
13
14     Thread t;
15     public Bienvenida() {
16
17         initComponents();
```

```

18     }
19
20
21     public void run() {
22         try{
23             this.setLocationRelativeTo(null);
24             this.setVisible(true);
25             Thread.sleep(2500);
26             this.dispose();
27
28             SimAS simas = new SimAS();
29             simas.lanzarEditor();
30         }catch(InterruptedException ex){
31             Logger.getLogger(Bienvenida.class.getName()).log(Level.
32                 SEVERE, null, ex);
33         }
34     }
35 }
36
37 /**
38  * This method is called from within the constructor to initialize
39  * the form.
40  * WARNING: Do NOT modify this code. The content of this method is
41  * always
42  * regenerated by the Form Editor.
43  */
44 @SuppressWarnings("unchecked")
45 // <editor-fold defaultstate="collapsed" desc="Generated Code">
46 // GEN-BEGIN: initComponents
47 private void initComponents() {
48
49     jLabel1 = new javax.swing.JLabel();
50     jLabel2 = new javax.swing.JLabel();
51     jLabel3 = new javax.swing.JLabel();
52     jLabel4 = new javax.swing.JLabel();
53     jLabel5 = new javax.swing.JLabel();
54     jLabel6 = new javax.swing.JLabel();
55     jLabel7 = new javax.swing.JLabel();
56
57     setDefaultCloseOperation(javax.swing.WindowConstants.
58         EXIT_ON_CLOSE);
59     setTitle("Bienvenido a SimAS");
60     setFocusTraversalPolicyProvider(true);
61     setUndecorated(true);
62
63     jLabel1.setIcon(new javax.swing.ImageIcon(getClass().
64         getResource("/es/uco/simas/resources/logo2Antes.png"))); //
65     NOI18N
66
67     jLabel2.setForeground(new java.awt.Color(33, 77, 72));
68     jLabel2.setText("Desarrollado por:");

```

```

64      jLabel3.setForeground(new java.awt.Color(33, 77, 72));
65      jLabel3.setText("Dirigido por: ");
66
67      jLabel4.setForeground(new java.awt.Color(63, 171, 160));
68      jLabel4.setText("Vanesa Gonzalez Perez");
69
70      jLabel5.setForeground(new java.awt.Color(63, 171, 160));
71      jLabel5.setText("Dr. Nicolas L. Fernandez Garcia");
72
73      jLabel6.setForeground(new java.awt.Color(33, 77, 72));
74      jLabel6.setText("Version: ");
75
76      jLabel7.setForeground(new java.awt.Color(63, 171, 160));
77      jLabel7.setText("1.0");
78
79      javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
80          getContentPane());
81      getContentPane().setLayout(layout);
82      layout.setHorizontalGroup(
83          layout.createParallelGroup(javax.swing.GroupLayout.
84              Alignment.LEADING)
85              .addGroup(layout.createSequentialGroup()
86                  .addGroup(layout.createParallelGroup(javax.
87                      swing.GroupLayout.Alignment.LEADING)
88                      .addGroup(layout.createSequentialGroup()
89                          .addComponent(jLabel2)
90                          .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
91                          .addComponent(jLabel4))
92                      .addGroup(layout.createSequentialGroup()
93                          .addComponent(jLabel3)
94                          .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
95                          .addComponent(jLabel5))
96                      .addGroup(layout.createSequentialGroup()
97                          .addComponent(jLabel6)
98                          .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
99                          .addComponent(jLabel7)))
100                  .addComponent(jLabel1))
101              .addGroup(layout.createSequentialGroup()
102                  .addGap(95, 95, 95)
103                  .addComponent(jLabel1)))
104          .addGap(41, 41, Short.MAX_VALUE));
105
106      layout.setVerticalGroup(
107          layout.createParallelGroup(javax.swing.GroupLayout.
108              Alignment.LEADING)
109              .addGroup(layout.createSequentialGroup()
110                  .addGap(21, 21, 21)

```

```

110         .addComponent(jLabel1)
111         .addGap(27, 27, 27)
112         .addGroup(layout.createParallelGroup(javax.swing.
            GroupLayout.Alignment.BASELINE)
113             .addComponent(jLabel2)
114             .addComponent(jLabel4))
115         .addGap(18, 18, 18)
116         .addGroup(layout.createParallelGroup(javax.swing.
            GroupLayout.Alignment.BASELINE)
117             .addComponent(jLabel3)
118             .addComponent(jLabel5))
119         .addGap(18, 18, 18)
120         .addGroup(layout.createParallelGroup(javax.swing.
            GroupLayout.Alignment.BASELINE)
121             .addComponent(jLabel6)
122             .addComponent(jLabel7))
123         .addContainerGap(25, Short.MAX_VALUE))
124     );
125
126     pack();
127 }// </editor-fold>//GEN-END: initComponents
128
129
130 // Variables declaration - do not modify//GEN-BEGIN: variables
131 private javax.swing.JLabel jLabel1;
132 private javax.swing.JLabel jLabel2;
133 private javax.swing.JLabel jLabel3;
134 private javax.swing.JLabel jLabel4;
135 private javax.swing.JLabel jLabel5;
136 private javax.swing.JLabel jLabel6;
137 private javax.swing.JLabel jLabel7;
138 // End of variables declaration//GEN-END: variables
139 }

```

### 2.2.2. SimAS.java

```

1 //SimAS
2
3 package es.uco.simas;
4
5 import es.uco.simas.editor.Editor;
6 import es.uco.simas.simulador.Simulador;
7 import es.uco.simas.util.gramatica.Gramatica;
8
9 /**
10  * @author vanesa
11  */
12 public class SimAS {
13     public SimAS() {
14

```

```

15     }
16
17     public void lanzarEditor(){ //Lanza el Editor
18         Editor editor = new Editor();
19         editor.setLocationRelativeTo(null);
20         editor.setVisible(true);
21     }
22
23     public void lanzarEditor(Gramatica gramatica){ //Editor con una
24         gramatica
25         Editor editor = new Editor(gramatica);
26         editor.setLocationRelativeTo(null);
27         editor.setVisible(true);
28     }
29
30     public void lanzarSimulador(int i, Gramatica gramatica, int met){
31         //Lanza el simulador
32         Simulador simulador = new Simulador(i, gramatica, met);
33         simulador.setLocationRelativeTo(null);
34         simulador.setVisible(true);
35     }
36     public void lanzarAyuda(){
37     }
38
39
40     public static void main(String args[]) {
41         /* Set the Nimbus look and feel */
42         //<editor-fold defaultstate="collapsed" desc=" Look and feel
43         setting code (optional) ">
44         /* If Nimbus (introduced in Java SE 6) is not available, stay
45         with the default look and feel.
46         * For details see http://download.oracle.com/javase/tutorial/
47         uiswing/lookandfeel/plaf.html
48         */
49
50         new Thread(new Bienvenida()).start();
51
52     try {
53         for (javax.swing.UIManager.LookAndFeelInfo info : javax.
54             swing.UIManager.getInstalledLookAndFeels()) {
55             if ("Nimbus".equals(info.getName())) {
56                 javax.swing.UIManager.setLookAndFeel(info.
57                     getClassName());
58                 break;
59             }
60         }
61     } catch (ClassNotFoundException ex) {
62         java.util.logging.Logger.getLogger(SimAS.class.getName()).
63             log(java.util.logging.Level.SEVERE, null, ex);
64     } catch (InstantiationException ex) {

```

```

60         java.util.logging.Logger.getLogger(SimAS.class.getName()).
           log(java.util.logging.Level.SEVERE, null, ex);
61     } catch (IllegalAccessException ex) {
62         java.util.logging.Logger.getLogger(SimAS.class.getName()).
           log(java.util.logging.Level.SEVERE, null, ex);
63     } catch (javax.swing.UnsupportedLookAndFeelException ex) {
64         java.util.logging.Logger.getLogger(SimAS.class.getName()).
           log(java.util.logging.Level.SEVERE, null, ex);
65     }
66     //</editor-fold>
67
68     /* Create and display the form */
69     java.awt.EventQueue.invokeLater(new Runnable() {
70         public void run() {
71
72             SimAS simas = new SimAS();
73         }
74     });
75
76 }
77 }

```

## 2.3. Paquete Centro de ayuda

### 2.3.1. AcercaDe.java

```

1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4  */
5  package es.uco.simas.centroayuda;
6
7  /**
8   *
9   * @author vanesa
10 */
11 public class AcercaDe extends javax.swing.JFrame {
12
13     /**
14      * Creates new form AcercaDe
15     */
16     public AcercaDe() {
17         initComponents();
18     }
19
20     /**
21      * This method is called from within the constructor to initialize
       the form.

```



```

22      * WARNING: Do NOT modify this code. The content of this method is
23      always
24      * regenerated by the Form Editor.
25      */
26      @SuppressWarnings("unchecked")
27      // <editor-fold defaultstate="collapsed" desc="Generated Code">//
28      GEN-BEGIN: initComponents
29      private void initComponents() {
30
31          jLabel1 = new javax.swing.JLabel();
32          jLabel2 = new javax.swing.JLabel();
33          jLabel3 = new javax.swing.JLabel();
34          jLabel4 = new javax.swing.JLabel();
35          jLabel5 = new javax.swing.JLabel();
36          jLabel6 = new javax.swing.JLabel();
37          jLabel7 = new javax.swing.JLabel();
38
39          setDefaultCloseOperation(javax.swing.WindowConstants.
40              EXIT_ON_CLOSE);
41
42          jLabel1.setIcon(new javax.swing.ImageIcon(getClass().
43              getResource("/es/uco/simas/resources/logo2Antes.png"))); //
44              NOI18N
45
46          jLabel2.setFont(new java.awt.Font("Ubuntu", 1, 15)); // NOI18N
47          jLabel2.setForeground(new java.awt.Color(33, 77, 72));
48          jLabel2.setText("Desarrollado por:");
49
50          jLabel3.setFont(new java.awt.Font("Ubuntu", 1, 15)); // NOI18N
51          jLabel3.setForeground(new java.awt.Color(63, 171, 160));
52          jLabel3.setText("Vanesa Gonz lez P rez");
53
54          jLabel4.setFont(new java.awt.Font("Ubuntu", 1, 15)); // NOI18N
55          jLabel4.setForeground(new java.awt.Color(33, 77, 72));
56          jLabel4.setText("Dirigido por:");
57
58          jLabel5.setFont(new java.awt.Font("Ubuntu", 1, 15)); // NOI18N
59          jLabel5.setForeground(new java.awt.Color(63, 171, 160));
60          jLabel5.setText("Dr. Nicol s L. Fern ndez Garc a");
61
62          jLabel6.setFont(new java.awt.Font("Ubuntu", 1, 15)); // NOI18N
63          jLabel6.setForeground(new java.awt.Color(33, 77, 72));
64          jLabel6.setText("Version: ");
65
66          jLabel7.setFont(new java.awt.Font("Ubuntu", 1, 15)); // NOI18N
67          jLabel7.setForeground(new java.awt.Color(63, 171, 160));
68          jLabel7.setText("1.0");
69
70          javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
71              getContentPane());
72          getContentPane().setLayout(layout);
73          layout.setHorizontalGroup(

```

```

68         layout.createParallelGroup(javax.swing.GroupLayout.
        Alignment.LEADING)
69     .addGroup(layout.createSequentialGroup()
70         .addGroup(layout.createParallelGroup(javax.swing.
        GroupLayout.Alignment.LEADING)
71             .addGroup(layout.createSequentialGroup()
72                 .addGap(92, 92, 92)
73                 .addComponent(jLabel1))
74             .addGroup(layout.createSequentialGroup()
75                 .addGap(37, 37, 37)
76                 .addGroup(layout.createParallelGroup(javax.
        swing.GroupLayout.Alignment.LEADING)
77                     .addGroup(layout.createSequentialGroup()
78                         .addComponent(jLabel4)
79                         .addGap(36, 36, 36)
80                         .addComponent(jLabel5))
81                     .addGroup(layout.createSequentialGroup()
82                         .addComponent(jLabel2)
83                         .addGap(29, 29, 29)
84                         .addComponent(jLabel3))
85                     .addGroup(layout.createSequentialGroup()
86                         .addGap(101, 101, 101)
87                         .addComponent(jLabel6)
88                         .addGap(18, 18, 18)
89                         .addComponent(jLabel7))))))
90         .addGap(34, Short.MAX_VALUE))
91     );
92     layout.setVerticalGroup(
93         layout.createParallelGroup(javax.swing.GroupLayout.
        Alignment.LEADING)
94     .addGroup(layout.createSequentialGroup()
95         .addGap(25, 25, 25)
96         .addComponent(jLabel1)
97         .addGap(36, 36, 36)
98         .addGroup(layout.createParallelGroup(javax.swing.
        GroupLayout.Alignment.BASELINE)
99             .addComponent(jLabel2)
100             .addComponent(jLabel3))
101         .addGap(18, 18, 18)
102         .addGroup(layout.createParallelGroup(javax.swing.
        GroupLayout.Alignment.BASELINE)
103             .addComponent(jLabel4)
104             .addComponent(jLabel5))
105         .addGap(18, 18, 18)
106         .addGroup(layout.createParallelGroup(javax.swing.
        GroupLayout.Alignment.BASELINE)
107             .addComponent(jLabel6)
108             .addComponent(jLabel7))
109         .addGap(32, Short.MAX_VALUE))
110     );
111
112     pack();
113 } // </editor-fold> //GEN-END: initComponents

```

```

114
115  /**
116   * @param args the command line arguments
117   */
118  public static void main(String args[]) {
119      /* Set the Nimbus look and feel */
120      //<editor-fold defaultstate="collapsed" desc=" Look and feel
        setting code (optional) ">
121      /* If Nimbus (introduced in Java SE 6) is not available, stay
        with the default look and feel.
122      * For details see http://download.oracle.com/javase/tutorial/
        uiswing/lookandfeel/plaf.html
123      */
124      try {
125          for (javax.swing.UIManager.LookAndFeelInfo info : javax.
                swing.UIManager.getInstalledLookAndFeels()) {
126              if ("Nimbus".equals(info.getName())) {
127                  javax.swing.UIManager.setLookAndFeel(info.
                        getClassName());
128                  break;
129              }
130          }
131      } catch (ClassNotFoundException ex) {
132          java.util.logging.Logger.getLogger(AcercaDe.class.getName())
                .log(java.util.logging.Level.SEVERE, null, ex);
133      } catch (InstantiationException ex) {
134          java.util.logging.Logger.getLogger(AcercaDe.class.getName())
                .log(java.util.logging.Level.SEVERE, null, ex);
135      } catch (IllegalAccessException ex) {
136          java.util.logging.Logger.getLogger(AcercaDe.class.getName())
                .log(java.util.logging.Level.SEVERE, null, ex);
137      } catch (javax.swing.UnsupportedLookAndFeelException ex) {
138          java.util.logging.Logger.getLogger(AcercaDe.class.getName())
                .log(java.util.logging.Level.SEVERE, null, ex);
139      }
140      //</editor-fold>
141
142      /* Create and display the form */
143      java.awt.EventQueue.invokeLater(new Runnable() {
144          public void run() {
145              new AcercaDe().setVisible(true);
146          }
147      });
148  }
149  // Variables declaration - do not modify//GEN-BEGIN:variables
150  private javax.swing.JLabel jLabel1;
151  private javax.swing.JLabel jLabel2;
152  private javax.swing.JLabel jLabel3;
153  private javax.swing.JLabel jLabel4;
154  private javax.swing.JLabel jLabel5;
155  private javax.swing.JLabel jLabel6;
156  private javax.swing.JLabel jLabel7;
157  // End of variables declaration//GEN-END:variables

```

158 }

## 2.4. Paquete Editor

### 2.4.1. ColCanLALR.java

```

1 //SimAS / Editor
2 // ColCanLALR
3
4 package es.uco.simas.editor;
5
6 import es.uco.simas.util.gramatica.Gramatica;
7 import es.uco.simas.util.gramatica.Simbolo;
8 import java.util.ArrayList;
9
10 /**
11  * @author vanesa
12  */
13 public class ColCanLALR {
14     Gramatica gramatica;
15     ArrayList<ConjElementosLR1> conjElementosLALR = new ArrayList();
16     ArrayList<ConjElementosLR1> conj2 = new ArrayList();
17     StringBuffer coleccion = new StringBuffer();
18     ColCanLR1 colLR1 = new ColCanLR1();
19     int i;
20     int v;
21     public ColCanLALR() { //Constructor vacio
22
23     }
24
25     public ColCanLALR(Gramatica gramatica, ColCanLR1 col){
26         this.gramatica = gramatica;
27         this.colLR1 = col;
28     }
29
30     public void construir() { //Construir la coleccion
31         ArrayList<ConjElementosLR1> conjLR1 = this.colLR1.
32             getConjElementosLR1();
33         unificarConjuntos();
34
35         ArrayList<ConjElementosLR1> c = colLR1.getConj2();
36         int w=0;
37         while(w < c.size()){
38             int z=0;
39             while(z < this.conjElementosLALR.size()){
40                 if(c.get(w).getI() == this.conjElementosLALR.get(z).
41                     getI()){

```

```

40         c.get(w).setV(this.conjElementosLALR.get(z).getV());
41         ;
42         conj2.add(c.get(w));
43     }
44     z++;
45 }
46 w++;
47 }
48
49 int k=0;
50
51 while(k < this.conjElementosLALR.size()){
52     this.conjElementosLALR.get(k).setY(k);
53     coleccion.append("I").append(this.conjElementosLALR.get(k).
54         getI());
55     if(this.conjElementosLALR.get(k).getV() != -1){
56         coleccion.append("-").append(this.conjElementosLALR.get
57             (k).getV());
58     }
59     coleccion.append(" = ");
60     imprimirConjunto(k);
61     coleccion.append("\n\n");
62     k++;
63 }
64
65 public String getColeccion(){ //Devuelve la coleccion
66     return this.coleccion.toString();
67 }
68
69 public void unificarConjuntos(){ //Unifica los conjuntos
70     ArrayList<ConjElementosLR1> conjLR1 = this.colLR1.
71         getConjElementosLR1();
72     int x=0;
73     while(x < conjLR1.size()){
74         int prueba = comprobarConjunto(conjLR1.get(x));
75         if(prueba == -1){
76             this.conjElementosLALR.add(conjLR1.get(x));
77         }else{
78             anticipacion(prueba, x);
79         }
80         x++;
81     }
82 }
83
84 public int comprobarConjunto(ConjElementosLR1 conj){
85     int conjunto = -1;
86     int i = 1;
87     while(i < this.conjElementosLALR.size()){
88         int j=0;
89         int iguales = 0;
90         while(j < conj.getElementosLR1().size()){

```

```

89         if (elementosIguales(conj.getElementosLR1().get(j), this
90             .conjElementosLALR.get(i).getElementosLR1().get(j))
91             ==1){
92             j++;
93             iguales = 1;
94         }else{
95             iguales =0;
96             break;
97         }
98     }
99     if(iguales == 1){
100         conjunto = i;
101         this.conjElementosLALR.get(i).setV(conj.getI());
102         ConjElementosLR1 el = conj;
103         el.setI(this.conjElementosLALR.get(i).getI());
104         el.setV(this.conjElementosLALR.get(i).getV());
105         this.conj2.add(el);
106         break;
107     }else
108         i++;
109 }
110 return conjunto;
111 }
112
113 public int elementosIguales(ElementosLR1 el1, ElementosLR1 el2){
114     int i=0;
115     int iguales =-1;
116
117     if(el1.getPosicion() == el2.getPosicion() && el1.getProduc
118         ().getAntec().getSimboloNT().getNombre().equals(el2.
119         getProduc().getAntec().getSimboloNT().getNombre())){
120         while(i < el1.getProduc().getConsec().size()){
121             iguales = 0;
122             if(el1.getProduc().getConsec().get(i).getNombre().
123                 equals(el2.getProduc().getConsec().get(i).
124                 getNombre())){
125                 i++;
126             }else{
127                 iguales =0;
128                 break;
129             }
130             iguales = 1;
131         }
132     }
133     return iguales;
134 }
135
136 void anticipacion(int conjunto, int igual){
137     ConjElementosLR1 conj = this.conjElementosLALR.get(conjunto);
138     ConjElementosLR1 conjIgual = this.colLR1.getConjElementosLR1().
139         get(igual);
140     ArrayList<String> anticipacion1 = conj.getElementosLR1().get(0)
141         .getAnticipacion();

```

```

134     ArrayList<String> anticipacion2 = conjIgual.getElementosLR1().
        get(0).getAnticipacion();
135     ArrayList<String> anticipacion = anticipacion1;
136     int k=0;
137
138     while(k < anticipacion2.size()){
139         int l=0;
140         int encontrado = -1;
141         while(l < anticipacion.size()){
142             if(anticipacion.get(l).equals(anticipacion2.get(k)))
143                 encontrado = 1;
144             l++;
145         }
146         if(encontrado == -1){
147             anticipacion.add(anticipacion2.get(k));
148         }
149         k++;
150     }
151     k=0;
152     while(k < this.conjElementosLALR.get(conjunto).getElementosLR1
        ().size()){
153         this.conjElementosLALR.get(conjunto).getElementosLR1().get(
            k).setAnticipacion(anticipacion);
154         k++;
155     }
156 }
157
158 void imprimirConjunto(int conj){
159     int j=0;
160     ArrayList<ElementosLR1> elementos = this.conjElementosLALR.get(
        conj).getElementosLR1();
161     coleccion.append(" { ");
162
163     while(j < elementos.size()){
164         coleccion.append(" [");
165         ArrayList<Simbolo> consec= elementos.get(j).getProduc().
            getConsec();
166         int posicion = elementos.get(j).getPosicion();
167         int k=0;
168         coleccion.append(elementos.get(j).getProduc().getAntec().
            getSimboloNT().getNombre());
169         coleccion.append(" \u2192 ");
170         if(posicion==0){
171             coleccion.append(" \u25CF ");
172         }
173         if("\u03B5".equals(consec.get(0).getNombre())){ //Si el
            consecuente contiene epsilon solo se imprime el punto
174             coleccion.append(" \u25CF ");
175         }else{
176             while(k < consec.size() ){
177                 coleccion.append(consec.get(k).getNombre()).append(
                    " ");
178                 k++;

```

```

179         if(posicion == k){
180             coleccion.append(" \u25CF ");
181         }
182     }
183 }
184 int l=0;
185 coleccion.append(", ");
186 while(l < elementos.get(j).getAnticipacion().size()){
187     coleccion.append(elementos.get(j).getAnticipacion()
188         .get(l));
189     l++;
190     if(l < elementos.get(j).getAnticipacion().size())
191         coleccion.append(", ");
192 }
193 coleccion.append("]");
194 j++;
195 if(j == elementos.size()){
196     coleccion.append("}");
197 }else
198     coleccion.append(", ");
199 }
200 }
201 public ColCanLR1 getColLR1() {
202     return colLR1;
203 }
204
205 public ArrayList<ConjElementosLR1> getConj2() {
206     return conj2;
207 }
208
209 public ArrayList<ConjElementosLR1> getConjElementosLALR() {
210     return this.conjElementosLALR;
211 }
212 }

```

### 2.4.2. ColCanLR0.java

```

1 //SimAS / Editor
2 // ColCanLR0
3
4 package es.uco.simas.editor;
5
6 import es.uco.simas.util.gramatica.Antecedente;
7 import es.uco.simas.util.gramatica.Consecuente;
8 import es.uco.simas.util.gramatica.Gramatica;
9 import es.uco.simas.util.gramatica.NoTerminal;
10 import es.uco.simas.util.gramatica.Produccion;
11 import es.uco.simas.util.gramatica.Simbolo;
12 import java.util.ArrayList;

```



```

13 |
14 | /**
15 |  * @author vanesa
16 |  */
17 | public class ColCanLR0 {
18 |     ArrayList<ConjElementosLR0> conjElementosLR0 = new ArrayList();
19 |     ArrayList<ConjElementosLR0> conj2 = new ArrayList();
20 |     Gramatica gramatica;
21 |     int i;
22 |     StringBuffer coleccion = new StringBuffer();
23 |
24 |     //\u00b7 punto
25 |     public ColCanLR0() { //Constructor vacio
26 |
27 |     }
28 |
29 |     public ColCanLR0(Gramatica gramatica) {
30 |         this.gramatica = gramatica;
31 |         ArrayList<Produccion> pr = this.gramatica.getPr();
32 |         if (!pr.get(0).getAntec().getSimboloNT().getNombre().equals(this
33 |             .gramatica.getSimbInicial()+" ")) {
34 |             Antecedente ant = new Antecedente();
35 |             ant.setSimboloNT(new NoTerminal(gramatica.getSimbInicial()+
36 |                 " ", gramatica.getSimbInicial()+" "));
37 |             Consecuente con = new Consecuente();
38 |             ArrayList<Simbolo> simb = new ArrayList();
39 |             simb.add(new Simbolo(gramatica.getSimbInicial(), gramatica.
40 |                 getSimbInicial()));
41 |             con.setConjSimbolos(simb);
42 |             Produccion produc = new Produccion();
43 |             produc.setConsec(simb);
44 |             produc.setAntec(ant);
45 |             pr.add(0, produc);
46 |             gramatica.setPr(pr);
47 |             this.gramatica = gramatica;
48 |         }
49 |     }
50 |
51 |     public void construir() {
52 |         ArrayList<String> simbolos = new ArrayList();
53 |         this.i=0;
54 |         int j=0;
55 |
56 |         coleccion.append("I").append(this.i).append(" = ");
57 |         this.conjElementosLR0.add(new ConjElementosLR0(this.gramatica))
58 |             ;
59 |
60 |         ArrayList<ElementosLR0> elementos = this.conjElementosLR0.get(
61 |             this.i).getElementosLR0();
62 |         while(j < elementos.size()){
63 |             if(elementos.get(j).getPosicion() < elementos.get(j).
64 |                 getProduc().getConsec().size()){
65 |                 if(simbolos.size() ==0)

```



```

109         k++;
110     }
111     if(encontrado == 0){
112         simbolos.add(elementos.get(j).getPivote());
113     }
114 }
115 }
116 j++;
117 }
118
119 if(simbolos.size()==0){
120     coleccion.append("\n\n\u2200 X \u2208 V: Ir_a (I"+
        this.conjElementosLR0.get(m).getI()+", X) = \u2205"
        );
121 }
122 j=0;
123 while(j < simbolos.size()){
124     this.i++;
125     int imprimir=-1;
126     ConjElementosLR0 conj = new ConjElementosLR0(i,m,
        simbolos.get(j), conjElementosLR0.get(m),this.
        gramatica);
127
128     int prueba = this.comprobarConjunto(conj);
129     if(prueba == -1){
130         this.conjElementosLR0.add(conj);
131         imprimir = this.i;
132     }else{
133         conj = new ConjElementosLR0(prueba,m,simbolos.get(j)
        ));
134         this.conj2.add(conj);
135         // conj = this.conjElementosLR0.get(prueba);
136         conj.setI(this.conjElementosLR0.get(prueba).getI())
        ;
137         imprimir = prueba;
138         this.i--;
139     }
140     j++;
141     coleccion.append("\n\n");
142     coleccion.append("Ir_a (I"+conj.col+", "+conj.simbolo+"
        ) = ");
143     imprimirConjunto(imprimir);
144     coleccion.append(" = I"+ conj.i);
145 }
146 m++;
147 }
148 }
149
150 void imprimirConjunto(int conj){
151     int j=0;
152     ArrayList<ElementosLR0> elementos = this.conjElementosLR0.get(
        conj).getElementosLR0();
153     coleccion.append("{ ");

```

```

154
155     while(j < elementos.size()){
156         ArrayList<Simbolo> consec= elementos.get(j).getProduc().
            getConsec();
157         int posicion = elementos.get(j).getPosicion();
158         int k=0;
159         coleccion.append(elementos.get(j).getProduc().getAntec().
            getSimboloNT().getNombre());
160         coleccion.append(" \u2192 ");
161         if(posicion==0){
162             coleccion.append(" \u25CF "); //\u25AA ");
163         }
164         if("\u03B5".equals(consec.get(0).getNombre())){ //Si el
            consecuente contiene epsilon solo se imprime el punto
165             coleccion.append(" \u25CF "); //\u25AA ");
166         }else{
167             while(k < consec.size() ){
168                 coleccion.append(consec.get(k).getNombre()).append(
169                     " ");
170                 k++;
171                 if(posicion == k){
172                     coleccion.append(" \u25CF "); //\u25AA "); //\u25CF "); //\u00b7 ");
173                 }
174             }
175             j++;
176             if(j == elementos.size())
177                 coleccion.append(" }");
178             else
179                 coleccion.append(" , ");
180         }
181     }
182
183     public String getColeccion(){
184         return this.coleccion.toString();
185     }
186
187     public int comprobarConjunto(ConjElementosLR0 conj){
188         int conjunto = -1;
189         int i = 1;
190
191         while(i < this.conjElementosLR0.size()){
192             // System.out.println(this.conjElementosLR0.get(i).
                getElementosLR0().get(0).getProduc().getConsec().get(0).
                getNombre());
193             int j=0;
194             int iguales = 0;
195             if(conj.getElementosLR0().size() == this.conjElementosLR0.
                get(i).getElementosLR0().size()){
196                 while(j < conj.getElementosLR0().size()){
197

```

```

198         if (elementosIguales (conj.getElementosLR0().get(j),
199             this.conjElementosLR0.get(i).getElementosLR0().
200                 get(j))==1){
201             j++;
202             iguales = 1;
203         }else{
204             iguales =0;
205             break;
206         }
207     }
208     if (iguales == 1){
209         conjunto = i;
210         break;
211     }else
212         i++;
213     }
214     return conjunto;
215 }
216 public int elementosIguales (ElementosLR0 el1 , ElementosLR0 el2){
217     int i=0;
218     int iguales =-1;
219
220     if (el1.getPosicion() == el2.getPosicion() && el1.getProduc().
221         getAntec().getSimboloNT().getNombre().equals (el2.getProduc()
222             .getAntec().getSimboloNT().getNombre())){
223         while (i < el1.getProduc().getConsec().size()){
224             iguales = 0;
225             if (el1.getProduc().getConsec().get(i).getNombre().
226                 equals (el2.getProduc().getConsec().get(i).getNombre()
227                     )){
228                 i++;
229             }else{
230                 iguales =0;
231                 break;
232             }
233             iguales = 1;
234         }
235     }
236     return iguales;
237 }
238
239 public ArrayList<ConjElementosLR0> getConjElementosLR0 () {
240     return conjElementosLR0;
241 }
242
243 public ArrayList<ConjElementosLR0> getConj2 () {
244     return conj2;
245 }

```

### 2.4.3. ColCanLR1.java

```

1 //SimAS / Editor
2 // ColCanLR1
3
4 package es.uco.simas.editor;
5
6 import es.uco.simas.util.gramatica.Antecedente;
7 import es.uco.simas.util.gramatica.Consecuente;
8 import es.uco.simas.util.gramatica.Gramatica;
9 import es.uco.simas.util.gramatica.NoTerminal;
10 import es.uco.simas.util.gramatica.Produccion;
11 import es.uco.simas.util.gramatica.Simbolo;
12 import java.util.ArrayList;
13
14 /**
15  * @author vanesa
16  */
17 public class ColCanLR1 {
18     Gramatica gramatica;
19     ArrayList<ConjElementosLR1> conjElementosLR1 = new ArrayList();
20     ArrayList<ConjElementosLR1> conj2 = new ArrayList();
21     StringBuffer coleccion = new StringBuffer();
22     int i;
23
24     public ColCanLR1() { //Constructor vacio
25
26     }
27     public ColCanLR1(Gramatica gramatica) {
28         this.gramatica = gramatica;
29
30         ArrayList<Produccion> pr = this.gramatica.getPr();
31         if (!pr.get(0).getAntec().getSimboloNT().getNombre().equals(this
32             .gramatica.getSimbInicial()+"'")) {
33             Antecedente ant = new Antecedente();
34             ant.setSimboloNT(new NoTerminal(gramatica.getSimbInicial()+
35                 "' ", gramatica.getSimbInicial()+"'"));
36             Consecuente con = new Consecuente();
37             ArrayList<Simbolo> simb = new ArrayList();
38             simb.add(new Simbolo(gramatica.getSimbInicial(), gramatica.
39                 getSimbInicial()));
40             con.setConjSimbolos(simb);
41             Produccion produc = new Produccion();
42             produc.setConsec(simb);
43             produc.setAntec(ant);
44             pr.add(0, produc);
45             gramatica.setPr(pr);
46             this.gramatica = gramatica;
47         }
48     }
49
50     public void construir() {
51         ArrayList<String> simbolos = new ArrayList();

```

```

49     this.i=0;
50     int j=0;
51
52     coleccion.append("I").append(this.i).append(" = ");
53     this.conjElementosLR1.add(new ConjElementosLR1(this.gramatica))
54         ;
55
56     ArrayList<ElementosLR1> elementos = this.conjElementosLR1.get(
57         this.i).getElementosLR1();
58     while(j < elementos.size()){
59         if(elementos.get(j).getPosicion() < elementos.get(j).
60             getProduc().getConsec().size()){
61             if(simbolos.size() ==0)
62                 simbolos.add(elementos.get(j).getPivote());
63             else{
64                 int k=0;
65                 int encontrado = 0;
66                 while(k < simbolos.size()){
67                     if(simbolos.get(k).equals(elementos.get(j).
68                         getPivote())){
69                         encontrado = 1;
70                         break;
71                     }else
72                         k++;
73                 }
74                 if(encontrado == 0)
75                     simbolos.add(elementos.get(j).getPivote());
76             }
77         }
78         j++;
79     }
80     imprimirConjunto(0);
81
82     j=0;
83     while(j < simbolos.size()){
84         this.i++;
85         this.conjElementosLR1.add(new ConjElementosLR1(i,0,simbolos
86             .get(j), conjElementosLR1.get(0),this.gramatica));
87         j++;
88         coleccion.append("\n\n");
89         coleccion.append(" Ir_a (I"+this.conjElementosLR1.get(j).
90             col+", "+this.conjElementosLR1.get(j).simbolo+" ) = ");
91         imprimirConjunto(j);
92         coleccion.append(" = I"+ this.conjElementosLR1.get(j).i);
93     }
94
95     int tam = this.conjElementosLR1.size();
96
97     int m =1;
98     while(m < this.conjElementosLR1.size()){
99         elementos = this.conjElementosLR1.get(m).getElementosLR1();
100        simbolos = new ArrayList();
101        j=0;

```

```

96     while(j < elementos.size()){
97         int posicion = elementos.get(j).getPosicion();
98         if(elementos.get(j).getPivote() != ""){
99             if(simbolos.size() == 0 )
100                 simbolos.add(elementos.get(j).getPivote());
101             else{
102                 int k=0;
103                 int encontrado = 0;
104                 while(k < simbolos.size()){
105                     if(simbolos.get(k).equals(elementos.get(j).
106                         getPivote())){
107                         encontrado = 1;
108                         break;
109                     }else
110                         k++;
111                 }
112                 if(encontrado ==0)
113                     simbolos.add(elementos.get(j).getPivote());
114             }
115             j++;
116         }
117     }
118     if(simbolos.size()==0){
119         coleccion.append("\n\n\u2200 X \u2208 V:      Ir_a (I"+
120             "this.conjElementosLR1.get(m).getI()+", X) = \u2205"
121             ");
122     }
123     j=0;
124     while(j < simbolos.size()){
125         this.i++;
126         int imprimir=-1;
127         ConjElementosLR1 conj = new ConjElementosLR1(i,m,
128             simbolos.get(j), conjElementosLR1.get(m),this.
129             gramatica);
130         int prueba = this.comprobarConjunto(conj);
131         if(prueba == -1){
132             this.conjElementosLR1.add(conj);
133             imprimir = this.i;
134         }else{
135             conj = new ConjElementosLR1(prueba,m,simbolos.get(j)
136             ));
137             this.conj2.add(conj);
138             conj.setI(this.conjElementosLR1.get(prueba).getI())
139             ;
140             imprimir = prueba;
141             this.i--;
142         }
143         j++;
144         coleccion.append("\n\n");
145         coleccion.append(" Ir_a (I"+conj.col+"", "+conj.simbolo+"
146             ") = ");

```



```

141         imprimirConjunto(imprimir);
142         coleccion.append(" = I"+ conj.i);
143     }
144     m++;
145 }
146
147 }
148
149 void imprimirConjunto(int conj){
150     int j=0;
151     ArrayList<ElementosLR1> elementos = this.conjElementosLR1.get(
152         conj).getElementosLR1();
153     coleccion.append("{ ");
154     while(j < elementos.size()){
155         coleccion.append(" ");
156         ArrayList<Simbolo> consec= elementos.get(j).getProduc().
157             getConsec();
158         int posicion = elementos.get(j).getPosicion();
159         int k=0;
160         coleccion.append(elementos.get(j).getProduc().getAntec().
161             getSimboloNT().getNombre());
162         coleccion.append(" \u2192 ");
163         if(posicion==0){
164             coleccion.append(" \u25CF "); //\u25AA ");
165         }
166         if("\u03B5".equals(consec.get(0).getNombre())){ //Si el
167             consecuente contiene epsilon solo se imprime el punto
168             coleccion.append(" \u25CF "); //\u25AA ");
169         }else{
170             while(k < consec.size() ){
171                 coleccion.append(consec.get(k).getNombre()).append(
172                     " ");
173                 k++;
174                 if(posicion == k){
175                     coleccion.append(" \u25CF "); //\u25AA "); //\u25CF "); //\u00b7 ");
176                 }
177             }
178         }
179         int l=0;
180         coleccion.append(", ");
181         while(l < elementos.get(j).getAnticipacion().size()){
182             coleccion.append(elementos.get(j).getAnticipacion().
183                 get(l));
184             l++;
185             if(l < elementos.get(j).getAnticipacion().size())
186                 coleccion.append(", ");
187         }
188         coleccion.append("]");
189         j++;
190         if(j == elementos.size()){

```

```

187         coleccion.append(" }");
188     }else
189         coleccion.append(" , ");
190     }
191 }
192
193 public String getColeccion(){
194     return this.coleccion.toString();
195 }
196
197 public ArrayList<ConjElementosLR1> getConjElementosLR1() {
198     return conjElementosLR1;
199 }
200
201 public ArrayList<ConjElementosLR1> getConj2() {
202     return conj2;
203 }
204
205 public int comprobarConjunto(ConjElementosLR1 conj){
206     int conjunto = -1;
207     int i = 1;
208     while(i < this.conjElementosLR1.size()){
209         int j=0;
210         int iguales = 0;
211         while(j < conj.getElementosLR1().size()){
212             if(elementosIguales(conj.getElementosLR1().get(j), this
                .conjElementosLR1.get(i).getElementosLR1().get(j))
                ==1 && this.anticipacionIgual(conj.getElementosLR1
                ().get(j), this.conjElementosLR1.get(i).
                getElementosLR1().get(j))==1){
213                 j++;
214                 iguales = 1;
215             }else{
216                 iguales =0;
217                 break;
218             }
219         }
220         if(iguales == 1){
221             conjunto = i;
222             break;
223         }else
224             i++;
225     }
226     return conjunto;
227 }
228
229 public int elementosIguales(ElementosLR1 el1 , ElementosLR1 el2){
230     int i=0;
231     int iguales =-1;
232
233     if(el1.getPosicion() == el2.getPosicion() && el1.getProduc().
        getAntec().getSimboloNT().getNombre().equals(el2.getProduc
        ().getAntec().getSimboloNT().getNombre())){

```

```

234         while(i < el1.getProduc().getConsec().size()){
235             iguales = 0;
236             if(el1.getProduc().getConsec().get(i).getNombre().
                equals(el2.getProduc().getConsec().get(i).getNombre
                    ())) {
237                 i++;
238             } else {
239                 iguales = 0;
240                 break;
241             }
242             iguales = 1;
243         }
244     }
245     return iguales;
246 }
247
248 public int anticipacionIgual(ElementosLR1 el1, ElementosLR1 el2){
249     int iguales = -1;
250     int j=0;
251
252     if(el1.getAnticipacion().size() == el2.getAnticipacion().size()
        ){
253         while(j < el1.getAnticipacion().size()){
254             if(el1.getAnticipacion().get(j).equals(el2.
                getAnticipacion().get(j))) {
255                 j++;
256             } else {
257                 iguales = 0;
258                 break;
259             }
260             iguales = 1;
261         }
262     } else {
263         iguales = 0;
264     }
265     return iguales;
266 }
267 }

```

#### 2.4.4. ConjElementosLALR.java

```

1 // SimAs / Editor
2 // ConjElementosLALR
3
4 package es.uco.simas.editor;
5
6 import es.uco.simas.util.gramatica.Gramatica;
7 import es.uco.simas.util.gramatica.NoTerminal;
8 import es.uco.simas.util.gramatica.Produccion;
9 import es.uco.simas.util.gramatica.Simbolo;

```

```

10 import java.util.ArrayList;
11
12 /**
13  * @author vanesa
14  */
15 public class ConjElementosLALR {
16     ArrayList<ElementosLALR> elementosLALR = new ArrayList();
17     Gramatica gramatica;
18     int i; // numero de iteracion
19     int col; //coleccion con la que se calcula
20     String simbolo = "";
21     ConjElementosLALR conj;
22
23     public ConjElementosLALR(Gramatica gramatica){
24         this.gramatica = gramatica;
25         this.primer();
26     }
27
28     public ConjElementosLALR(int i, int col, String simbolo,
29         ConjElementosLALR conj, Gramatica gr){
30         this.i = i;
31         this.col = col;
32         this.simbolo = simbolo;
33         this.conj = conj;
34         this.gramatica = gr;
35         this.elementosLALR = new ArrayList();
36         this.construir();
37     }
38     public ConjElementosLALR(int i, int col, String simbolo){
39         this.i = i;
40         this.col = col;
41         this.simbolo = simbolo;
42     }
43
44     void primero(){
45         ArrayList<Produccion> producciones = this.gramatica.getPr();
46         ArrayList<String> simb = new ArrayList();
47         Produccion prod = producciones.get(0);
48         ArrayList<String> ant = new ArrayList();
49         ElementosLALR aux = new ElementosLALR();
50         ant.add("$");
51
52         //S' -> .S, $
53         this.elementosLALR.add(new ElementosLALR(prod,0,prod.getConsec
54             ().get(0).getNombre(), ant));
55         aux = this.elementosLALR.get(this.elementosLALR.size()-1);
56
57         String simbolo = prod.getConsec().get(0).getNombre();
58         int i=1;
59         while(i<producciones.size()){
60             ant = new ArrayList();
61             if(simbolo.equals(producciones.get(i).getAntec().
62                 getSimboloNT().getNombre())){

```

```

60         if (calcularAnt(aux) != null) {
61             ant = calcularAnt(aux);
62         } else {
63             ant = aux.getAnticipacion();
64         }
65         this.elementosLALR.add(new ElementosLALR(producciones.
            get(i), 0, producciones.get(i).getConsec().get(0).
            getNombre(), ant));
66     }
67     i++;
68 }
69
70 i=1;
71 int tam = this.elementosLALR.size();
72 while (i < tam) {
73     if (this.elementosLALR.get(i).getPosicion() != this.
        elementosLALR.get(i).getProduc().getConsec().size()) {
74         if (this.gramatica.isNoTerminal(this.elementosLALR.get(i)
            .getProduc().getConsec().get(this.elementosLALR.
            get(i).getPosicion()).getNombre())) {
75             int j=0;
76             simbolo = this.elementosLALR.get(i).getProduc().
                getConsec().get(this.elementosLALR.get(i).
                getPosicion()).getNombre();
77             aux = this.elementosLALR.get(i);
78             while (j < producciones.size()) {
79                 if (simbolo.equals(producciones.get(j).getAntec
                    ().getSimboloNT().getNombre())) {
80                     ant = new ArrayList();
81                     if (calcularAnt(aux) != null) {
82                         ant = calcularAnt(aux);
83                     } else {
84                         ant = aux.getAnticipacion();
85                     }
86                     this.elementosLALR.add(new ElementosLALR(
                        producciones.get(j), 0, producciones.get
                        (j).getConsec().get(0).getNombre(), ant
                        ));
87                 }
88                 j++;
89             }
90         }
91     }
92     i++;
93 }
94 if (tam != this.elementosLALR.size()) {
95     do {
96         i=tam;
97         tam = this.elementosLALR.size();
98         while (i < tam) {
99             if (this.elementosLALR.get(i).getPosicion() != this.
                elementosLALR.get(i).getProduc().getConsec().
                size()) {

```

```

100         if (this.gramatica.isNoTerminal(this.
101             elementosLALR.get(i).getProduc().getConsec
102             ().get(this.elementosLALR.get(i).
103                 getPosicion()).getNombre())){
104             int j=0;
105             simbolo = this.elementosLALR.get(i).
106                 getProduc().getConsec().get(this.
107                     elementosLALR.get(i).getPosicion()).
108                     getNombre();
109             aux = this.elementosLALR.get(i);
110             while(j < producciones.size()){
111                 if(simbolo.equals(producciones.get(j).
112                     getAntec().getSimboloNT().getNombre
113                     ()))){
114                     ant = new ArrayList();
115                     if(calcularAnt(aux)!=null){
116                         ant = calcularAnt(aux);
117                     }else{
118                         ant = aux.getAnticipacion();
119                     }
120                     this.elementosLALR.add(new
121                         ElementosLALR(producciones.get(
122                             j),0, producciones.get(j).
123                             getConsec().get(0).getNombre(),
124                             ant));
125                 }
126                 j++;
127             }
128         }
129         i++;
130     }
131     while(tam > this.elementosLALR.size());
132     this.agrupamos(this.elementosLALR);
133 }
134
135 void construir(){
136     ArrayList<ElementosLALR> elementos = this.conj.getElementosLALR
137         ();
138     ArrayList<String> simb = new ArrayList();
139     ArrayList<Produccion> producciones = new ArrayList();
140     producciones = this.gramatica.getPr();
141     String simbolo = new String();
142     ArrayList<String> ant = new ArrayList();
143     ElementosLALR aux = new ElementosLALR();
144     ArrayList<ElementosLALR> elem = new ArrayList();
145
146     int j=0;
147     while(j<elementos.size()){
148         ElementosLALR el = new ElementosLALR(elementos.get(j).
149             getProduc(),elementos.get(j).getPosicion(),elementos.
150             get(j).getPivote(), elementos.get(j).getAnticipacion())

```

```

138         ;
139         if (elementos.get(j).getProduc().getConsec().get(0).
140             getNombre().equals("\u03b5")){
141             el.setPosicion(el.getPosicion()+1);
142             el.setPivote("");
143         }
144         if (el.getPivote().equals(this.simbolo)){
145
146             el.setPosicion(el.getPosicion()+1);
147
148             String pivote=el.getPivote();
149             ArrayList<Simbolo> consec = el.getProduc().getConsec();
150             int k=0;
151             simb = new ArrayList();
152
153             if (el.posicion < consec.size()){ //Seleccionar pivote
154                 if (consec.get(el.posicion-1).getNombre().equals(
155                     pivote) ){
156
157                     el.setPivote(consec.get(el.getPosicion()).
158                         getNombre());
159
160                     int m =0;
161                     if (simb.size()==0){
162                         simb.add(el.getPivote());
163                     }else{
164                         int encontrado = 0;
165                         while(m< simb.size()){
166                             if (simb.get(m).equals(el.getPivote())){
167                                 encontrado = 1;
168                                 break;
169                             }else
170                                 m++;
171                         }
172                         if (encontrado == 0){
173                             simb.add(el.getPivote());
174                         }
175                     }
176                     this.elementosLALR.add(new ElementosLALR(el.
177                         getProduc(),el.getPosicion(), el.getPivote
178                         (), el.getAnticipacion()));
179                     if (simb.size() !=0)
180                         elem.add(this.elementosLALR.get(this.
181                             elementosLALR.size()-1));
182                 }
183             }else{
184                 el.setPivote("");
185                 this.elementosLALR.add(new ElementosLALR(el.
186                     getProduc(),el.getPosicion(), el.getPivote(),el
187                     .getAnticipacion()));
188             }
189         }
190     }
191     j++;

```

```

182     }
183
184     int k =0;
185     if (simb.size() != 0){
186         while(k < simb.size()){
187             simbolo = simb.get(k);
188             aux = elem.get(k);
189             ant = new ArrayList();
190             ant = new ArrayList();
191             if (calcularAnt(aux) != null){
192                 ant = calcularAnt(aux);
193             } else{
194                 ant = aux.getAnticipacion();
195             }
196             int i=1;
197             while(i < producciones.size()){
198
199                 if (simbolo.equals(producciones.get(i).getAntec().
200                     getSimboloNT().getNombre())){
201                     if (producciones.get(i).getConsec().get(0).
202                         getNombre().equals("\u03b5")){
203                         this.elementosLALR.add(new ElementosLALR(
204                             producciones.get(i),1, "", ant));
205                     } else{
206                         this.elementosLALR.add(new ElementosLALR(
207                             producciones.get(i),0, producciones.get
208                                 (i).getConsec().get(0).getNombre(), ant)
209                             );
210                     }
211                     elem.add(this.elementosLALR.get(this.
212                         elementosLALR.size()-1));
213                     int l=0;
214                     while(l < simb.size()){
215                         if (!simb.get(k).equals(producciones.get(i).
216                             getConsec().get(0).getNombre())){
217                             simb.add(producciones.get(i).getConsec
218                                 ().get(0).getNombre());
219                             break;
220                         }
221                         l++;
222                     }
223                     i++;
224                 }
225                 k++;
226             }
227         }
228     }
229 }
230
231 public ArrayList<String> calcularAnt(ElementosLALR elemento){
232     ArrayList<String> ant = new ArrayList();
233     ArrayList<Simbolo> consec = elemento.getProduc().getConsec();
234     ArrayList<NoTerminal> nterm = this.gramatica.getNoTerm();

```



```

226     int posicion = elemento.getPosicion();
227     if(posicion < consec.size()-1){
228         Simbolo simbolo = consec.get(posicion+1);
229         if(this.gramatica.isTerminal(simbolo.getNombre())){
230             ant.add(simbolo.getNombre());
231             return ant;
232         }else{
233             int j=0;
234             while(j < nterm.size()){
235                 if(nterm.get(j).getNombre().equals(simbolo.
236                     getNombre())){
237                     int k=0;
238                     while(k < nterm.get(j).getPrimeros().size()){
239                         ant.add(nterm.get(j).getPrimeros().get(k).
240                             getNombre());
241                         k++;
242                     }
243                     break;
244                 }else{
245                     j++;
246                 }
247             }
248             return ant;
249         }
250     }else
251         return null;
252 }
253
254 public void agrupamos(ArrayList<ElementosLALR> elementos){
255     int i=1;
256     ArrayList<ElementosLALR> elementosLR = new ArrayList();
257     ElementosLALR iguales = new ElementosLALR();
258     elementosLR.add(elementos.get(0));
259     while(i < elementos.size()){
260         ElementosLALR el = elementos.get(i);
261         int j=0;
262         while(j < elementosLR.size()){
263             iguales = null;
264             if(elementosLR.get(j).getPosicion() == el.getPosicion()
265                 && el.getProduc().getConsec().size() ==
266                 elementosLR.get(j).getProduc().getConsec().size()){
267                 int x=0;
268                 int ig=0;
269                 while(x < el.getProduc().getConsec().size()){
270                     if(!elementosLR.get(j).getProduc().getConsec().
271                         get(x).getNombre().equals(el.getProduc().
272                             getConsec().get(x).getNombre())){
273                         break;
274                     }else{
275                         x++;
276                         ig=1;
277                     }
278                 }
279             }
280         }
281     }

```

```

273         if(ig==1)
274             iguales = elementosLR.get(j);
275     }
276     if(iguales == null)
277         j++;
278     else
279         break;
280 }
281 if(iguales == null){
282     elementosLR.add(el);
283 }else{
284     ArrayList<String> ant = new ArrayList();
285     int k=0;
286     while(k < el.getAnticipacion().size()){
287         ant.add(el.getAnticipacion().get(k));
288         k++;
289     }
290     int l=0;
291     while(l < iguales.getAnticipacion().size()){
292         ant.add(iguales.getAnticipacion().get(l));
293         l++;
294     }
295     int m=1;
296     ArrayList<String> ant2 = new ArrayList();
297     ant2.add(ant.get(0));
298     while(m < ant.size()){
299         int n=0;
300         int enc = 0;
301         while(n < ant2.size()){
302             if(ant.get(m).equals(ant2.get(n))){
303                 enc = 1;
304                 break;
305             }else
306                 n++;
307         }
308         if(enc == 0)
309             ant2.add(ant.get(m));
310         m++;
311     }
312     elementosLR.set(j, new ElementosLALR(el.getProduc(), el.
        getPosicion(), el.getPivote(), ant2));
313 }
314 i++;
315 }
316 this.setElementosLALR(elementosLR);
317 }
318
319 public ArrayList<ElementosLALR> getElementosLALR() {
320     return elementosLALR;
321 }
322
323 public void setElementosLALR(ArrayList<ElementosLALR> elementosLALR
    ) {

```

```
324         this.elementosLALR = elementosLALR;
325     }
326
327     public int getI() {
328         return i;
329     }
330
331     public void setI(int i) {
332         this.i = i;
333     }
334 }
```

### 2.4.5. ConjElementosLR0.java

```
1  // SimAs / Editor
2  // ConjElementosLR0
3
4  package es.uco.simas.editor;
5
6  import es.uco.simas.util.gramatica.Gramatica;
7  import es.uco.simas.util.gramatica.Produccion;
8  import es.uco.simas.util.gramatica.Simbolo;
9  import java.util.ArrayList;
10
11 /**
12  * @author vanesa
13  */
14 public class ConjElementosLR0 {
15     ArrayList<ElementosLR0> elementosLR0 = new ArrayList();
16     int i; // numero de iteracion
17     int col; //coleccion con la que se calcula
18     String simbolo = "";
19     Gramatica gramatica;
20     ConjElementosLR0 conj;
21
22     public ConjElementosLR0(Gramatica gramatica){
23         this.gramatica = gramatica;
24         this.primer();
25     }
26
27     public ConjElementosLR0(int i, int col, String simbolo,
28         ConjElementosLR0 conj, Gramatica gr){
29         this.i = i;
30         this.col = col;
31         this.simbolo = simbolo;
32         this.conj = conj;
33         this.gramatica = gr;
34         this.elementosLR0 = new ArrayList();
35         this.construir();
36     }
37 }
```

```

36
37 public ConjElementosLR0(int i, int col, String simbolo){
38     this.i = i;
39     this.col = col;
40     this.simbolo = simbolo;
41 }
42
43 void primero(){
44     ArrayList<Produccion> producciones = this.gramatica.getPr();
45     ArrayList<String> simb = new ArrayList();
46     Produccion prod = producciones.get(0);
47
48     this.elementosLR0.add(new ElementosLR0(prod,0,prod.getConsec().
49         get(0).getNombre()));
50
51     String simbolo = prod.getConsec().get(0).getNombre();
52     int i=1;
53     while(i<producciones.size()){
54         if(simbolo.equals(producciones.get(i).getAntec().
55             getSimboloNT().getNombre())){
56             if(producciones.get(i).getConsec().get(0).getNombre().
57                 equals("\u03b5")){
58                 this.elementosLR0.add(new ElementosLR0(producciones
59                     .get(i),1, ""));
60             }else{
61                 this.elementosLR0.add(new ElementosLR0(producciones
62                     .get(i),0, producciones.get(i).getConsec().get
63                     (0).getNombre()));
64             }
65         }
66         i++;
67     }
68     i=1;
69     while(i<this.elementosLR0.size()){
70         if(!this.elementosLR0.get(i).getPivote().equals(simbolo)){
71             int k =0;
72             if(simb.size()==0){
73                 simb.add(this.elementosLR0.get(i).getPivote());
74             }else{
75                 int encontrado = 0;
76                 while(k< simb.size()){
77                     if(simb.get(k).equals(this.elementosLR0.get(i).
78                         getPivote())){
79                         encontrado = 1;
80                         break;
81                     }else
82                         k++;
83                 }
84                 if(encontrado ==0){
85                     simb.add(this.elementosLR0.get(i).getPivote());
86                 }
87             }
88         }
89     }
90 }

```

```

82         i++;
83     }
84
85     int k =0;
86     if (simb.size() != 0){
87         while(k < simb.size()){
88             simbolo = simb.get(k);
89             i=1;
90             while(i<producciones.size()){
91                 if (simbolo.equals(producciones.get(i).getAntec().
92                     getSimboloNT().getNombre())){
93                     if (producciones.get(i).getConsec().get(0).
94                         getNombre().equals("\u03b5")){
95                         this.elementosLR0.add(new ElementosLR0(
96                             producciones.get(i),1, ""));
97                     }else{
98                         this.elementosLR0.add(new ElementosLR0(
99                             producciones.get(i),0, producciones.get
100                                 (i).getConsec().get(0).getNombre()));
101                     }
102                     int l=0;
103                     int encontrado =0;
104                     while(l < simb.size()){
105                         if (simb.get(l).equals(producciones.get(i).
106                             getConsec().get(0).getNombre())){
107                             encontrado = 1;
108                             break;
109                         }else{
110                             encontrado = 0;
111                             l++;
112                         }
113                     }
114                     if (encontrado == 0){
115                         simb.add(producciones.get(i).getConsec().
116                             get(0).getNombre());
117                     }
118                 }
119                 i++;
120             }
121             k++;
122         }
123     }
124 }
125
126 void construir(){
127     ArrayList<ElementosLR0> elementos = this.conj.getElementosLR0()
128     ;
129     ArrayList<String> simb = new ArrayList();
130     ArrayList<Produccion> producciones = new ArrayList();
131     producciones = this.gramatica.getPr();
132     String simbolo = new String();
133     int j=0;
134     while(j<elementos.size()){

```

```

127         ElementosLR0 el = new ElementosLR0(elementos.get(j).
            getProduc(), elementos.get(j).getPosicion(), elementos.
            get(j).getPivote());
128         if(elementos.get(j).getProduc().getConsec().get(0).
            getNombre().equals("\u03b5")){
129             el.setPosicion(el.getPosicion()+1);
130             el.setPivote("");
131         }
132
133         if(el.getPivote().equals(this.simbolo)){
134             el.setPosicion(el.getPosicion()+1);
135             String pivote = el.getPivote();
136             ArrayList<Simbolo> consec = el.getProduc().getConsec();
137             int k=0;
138
139             if(el.posicion < consec.size()){ //Seleccionar pivote
140                 if(consec.get(el.posicion-1).getNombre().equals(
                    pivote)){
141                     el.setPivote(consec.get(el.getPosicion()).
                        getNombre());
142                     int m=0;
143                     if(simb.size()==0){
144                         simb.add(el.getPivote());
145                     }else{
146                         int encontrado = 0;
147                         while(m< simb.size()){
148                             if(simb.get(m).equals(el.getPivote())){
149                                 encontrado = 1;
150                                 break;
151                             }else
152                                 m++;
153                         }
154                         if(encontrado == 0){
155                             simb.add(el.getPivote());
156                         }
157                     }
158
159                     this.elementosLR0.add(new ElementosLR0(el.
                        getProduc(), el.getPosicion(), el.getPivote
                        ()));
160                 }
161             }else{
162                 el.setPivote("");
163
164                 el.setPivote("");
165                 this.elementosLR0.add(new ElementosLR0(el.getProduc
                    (), el.getPosicion(), el.getPivote()));
166             }
167         }
168         j++;
169     }
170     int k =0;
171     if(simb.size() != 0){

```

```

172         while(k < simb.size()){
173             simbolo = simb.get(k);
174             int i=1;
175             while(i<producciones.size()){
176
177                 if(simbolo.equals(producciones.get(i).getAntec().
178                     getSimboloNT().getNombre())){
179                     if(producciones.get(i).getConsec().get(0).
180                         getNombre().equals("\u03b5")){
181                         this.elementosLR0.add(new ElementosLR0(
182                             producciones.get(i),1, ""));
183                     }else{
184                         this.elementosLR0.add(new ElementosLR0(
185                             producciones.get(i),0, producciones.get
186                             (i).getConsec().get(0).getNombre()));
187                     }
188
189                     int l=0;
190                     while(l < simb.size()){
191                         if(!simb.get(k).equals(producciones.get(i).
192                             getConsec().get(0).getNombre())){
193                             simb.add(producciones.get(i).getConsec
194                                 ().get(0).getNombre());
195                             break;
196                         }
197                         l++;
198                     }
199                 }
200             }
201             i++;
202         }
203         k++;
204     }
205 }
206
207 public ArrayList<ElementosLR0> getElementosLR0() {
208     return elementosLR0;
209 }
210
211 public void setElementosLR0(ArrayList<ElementosLR0> elementosLR0) {
212     this.elementosLR0 = elementosLR0;
213 }
214
215 public int getI() {
216     return i;
217 }
218
219 public void setI(int i) {
220     this.i = i;
221 }

```

### 2.4.6. ConjElementosLR1.java

```
1 // SimAs / Editor
2 // ConjElementosLR1
3
4 package es.uco.simas.editor;
5
6 import es.uco.simas.util.gramatica.Gramatica;
7 import es.uco.simas.util.gramatica.NoTerminal;
8 import es.uco.simas.util.gramatica.Produccion;
9 import es.uco.simas.util.gramatica.Simbolo;
10 import java.util.ArrayList;
11
12 /**
13  * @author vanesa
14  */
15 public class ConjElementosLR1 {
16     ArrayList<ElementosLR1> elementosLR1 = new ArrayList();
17     Gramatica gramatica;
18     int i; // numero de iteracion
19     int col; //coleccion con la que se calcula
20     String simbolo = "";
21     ConjElementosLR1 conj;
22     int v=-1;
23     int y;
24
25     public ConjElementosLR1() {
26
27     }
28
29     public ConjElementosLR1(Gramatica gramatica){
30         this.gramatica = gramatica;
31         this.primer();
32     }
33
34     public ConjElementosLR1(int i, int col, String simbolo,
35         ConjElementosLR1 conj, Gramatica gr){
36         this.i = i;
37         this.col = col;
38         this.simbolo = simbolo;
39         this.conj = conj;
40         this.gramatica = gr;
41         this.elementosLR1 = new ArrayList();
42         this.construir();
43     }
44     public ConjElementosLR1(int i, int col, String simbolo){
45         this.i = i;
46         this.col = col;
47         this.simbolo = simbolo;
48     }
49
50     void primer(){
51         ArrayList<Produccion> producciones = this.gramatica.getPr();
```



```

51     ArrayList<String> simb = new ArrayList();
52     Produccion prod = producciones.get(0);
53     ArrayList<String> ant = new ArrayList();
54     ElementosLR1 aux = new ElementosLR1();
55     ant.add("$");
56
57     //S' -> .S, $
58     this.elementosLR1.add(new ElementosLR1(prod,0,prod.getConsec().
59         get(0).getNombre(), ant));
60
61     aux = this.elementosLR1.get(this.elementosLR1.size()-1);
62
63     String simbolo = prod.getConsec().get(0).getNombre();
64     int i=1;
65     while(i<producciones.size()){
66         ant = new ArrayList();
67         if(simbolo.equals(producciones.get(i).getAntec().
68             getSimboloNT().getNombre())){
69             if(calcularAnt(aux)!=null){
70                 ant = calcularAnt(aux);
71             }else{
72                 ant = aux.getAnticipacion();
73             }
74             if(producciones.get(i).getConsec().get(0).getNombre().
75                 equals("\u03b5")){
76                 this.elementosLR1.add(new ElementosLR1(producciones
77                     .get(i),1, "",ant));
78             }else{
79                 this.elementosLR1.add(new ElementosLR1(producciones
80                     .get(i),0, producciones.get(i).getConsec().get
81                     (0).getNombre(),ant));
82             }
83         }
84         i++;
85     }
86     i=1;
87     int tam = this.elementosLR1.size();
88     while (i < tam){
89         if(this.elementosLR1.get(i).getPosicion() != this.
90             elementosLR1.get(i).getProduc().getConsec().size()){
91             if(this.gramatica.isNoTerminal(this.elementosLR1.get(i)
92                 .getProduc().getConsec().get(this.elementosLR1.get(i)
93                     .getPosicion()).getNombre())){
94                 int j=0;
95                 simbolo = this.elementosLR1.get(i).getProduc().
96                     getConsec().get(this.elementosLR1.get(i).
97                         getPosicion()).getNombre();
98                 aux = this.elementosLR1.get(i);
99                 while(j < producciones.size()){
100                     if(simbolo.equals(producciones.get(j).getAntec
101                         ().getSimboloNT().getNombre())){
102                         ant = new ArrayList();
103                         if(calcularAnt(aux)!=null){
104                             ant = calcularAnt(aux);

```

```

92         }else{
93             ant = aux.getAnticipacion();
94         }
95         if(producciones.get(j).getConsec().get(0).
           getNombre().equals("\u03b5")){
96             this.elementosLR1.add(new ElementosLR1(
           producciones.get(j),1, "", ant));
97         }else{
98             this.elementosLR1.add(new ElementosLR1(
           producciones.get(j),0, producciones
           .get(j).getConsec().get(0).
           getNombre(), ant));
99         }
100     }
101     j++;
102 }
103 }
104 }
105 i++;
106 }
107 if(tam != this.elementosLR1.size()){
108     do{
109         i=tam;
110         tam = this.elementosLR1.size();
111         while (i < tam){
112             if(this.elementosLR1.get(i).getPosicion() != this.
           elementosLR1.get(i).getProduc().getConsec().
           size()){
113                 if(this.gramatica.isNoTerminal(this.
           elementosLR1.get(i).getProduc().getConsec().
           get(this.elementosLR1.get(i).getPosicion().
           getNombre()))){
114                     int j=0;
115                     simbolo = this.elementosLR1.get(i).
           getProduc().getConsec().get(this.
           elementosLR1.get(i).getPosicion().
           getNombre());
116                     aux = this.elementosLR1.get(i);
117                     while(j < producciones.size()){
118                         if(simbolo.equals(producciones.get(j).
           getAntec().getSimboloNT().getNombre().
           ())) {
119                             ant = new ArrayList();
120                             if(calcularAnt(aux)!=null){
121                                 ant = calcularAnt(aux);
122                             }else{
123                                 ant = aux.getAnticipacion();
124                             }
125                             if(producciones.get(j).getConsec().
           get(0).getNombre().equals("\u03b5")){
126                                 this.elementosLR1.add(new
           ElementosLR1(producciones.

```

```

127         get(j),1, "", ant));
128     }else{
        this.elementosLR1.add(new
            ElementosLR1(producciones.
                get(j),0, producciones.get(
                    j).getConsec().get(0).
                        getNombre(), ant));
129     }
130     }
131     j++;
132     }
133     }
134     }
135     i++;
136     }
137     }while(tam > this.elementosLR1.size());
138 }
139 this.agrupamos(this.elementosLR1);
140 }
141
142 void construir(){
143     ArrayList<ElementosLR1> elementos = this.conj.getElementosLR1()
144     ;
145     ArrayList<String> simb = new ArrayList();
146     ArrayList<Produccion> producciones = new ArrayList();
147     producciones = this.gramatica.getPr();
148     String simbolo = new String();
149     ArrayList<String> ant = new ArrayList();
150     ElementosLR1 aux = new ElementosLR1();
151     ArrayList<ElementosLR1> elem = new ArrayList();
152
153     int j=0;
154     while(j<elementos.size()){
        ElementosLR1 el = new ElementosLR1(elementos.get(j).
            getProduc(),elementos.get(j).getPosicion(),elementos.
                get(j).getPivote(), elementos.get(j).getAnticipacion())
            ;
155         if(elementos.get(j).getProduc().getConsec().get(0).
            getNombre().equals("\u03b5")){
156             el.setPosicion(el.getPosicion()+1);
157             el.setPivote("");
158         }
159         if(el.getPivote().equals(this.simbolo)){
160
161             el.setPosicion(el.getPosicion()+1);
162
163             String pivote=el.getPivote();
164             ArrayList<Simbolo> consec = el.getProduc().getConsec();
165             int k=0;
166             simb = new ArrayList();
167
168             if(el.posicion < consec.size()){ //Seleccionar pivote

```

```

169         if (consec.get(el.posicion-1).getNombre().equals(
170             pivote) ){
171             el.setPivote(consec.get(el.getPosicion()).
172                 getNombre());
173
174             int m=0;
175             if(simb.size()==0){
176                 simb.add(el.getPivote());
177             }else{
178                 int encontrado = 0;
179                 while(m< simb.size()){
180                     if(simb.get(m).equals(el.getPivote())){
181                         encontrado = 1;
182                         break;
183                     }else
184                         m++;
185                 }
186                 if(encontrado == 0){
187                     simb.add(el.getPivote());
188                 }
189             }
190             this.elementosLR1.add(new ElementosLR1(el.
191                 getProduc(),el.getPosicion(), el.getPivote
192                 (), el.getAnticipacion()));
193             if(simb.size() !=0)
194                 elem.add(this.elementosLR1.get(this.
195                     elementosLR1.size()-1));
196         }
197     }
198     j++;
199 }
200     int prueba=0;
201     int k =0;
202     if(simb.size() != 0){
203         while(k < simb.size()){
204             simbolo = simb.get(k);
205             aux = elem.get(k);
206             ant = new ArrayList();
207             ArrayList<String> ant2 = new ArrayList();
208
209             if(calcularAnt(aux)!=null){
210                 ant = calcularAnt(aux);
211             }else{
212                 ant = aux.getAnticipacion();
213             }
214             int i=1;

```

```

215         while(i<producciones.size()){
216             if(simbolo.equals(producciones.get(i).getAntec().
                getSimboloNT().getNombre())){
217                 if(producciones.get(i).getConsec().get(0).
                    getNombre().equals("\u03b5")){
218                     this.elementosLR1.add(new ElementosLR1(
                        producciones.get(i),1,"",ant));
219                 }else{
220                     this.elementosLR1.add(new ElementosLR1(
                        producciones.get(i),0,producciones.get
                        (i).getConsec().get(0).getNombre(),ant)
                    );
221                 }
222                 if(this.gramatica.isNoTerminal(producciones.get
                    (i).getConsec().get(0).getNombre())){
223                     if(prueba <1){
224                         elem.add(this.elementosLR1.get(this.
                            elementosLR1.size()-1));
225                         simb.add(producciones.get(i).getConsec
                            ().get(0).getNombre());
226                         prueba++;
227                     }
228                 }
229             }
230             i++;
231         }
232         k++;
233     }
234 }
235 this.agrupamos(this.elementosLR1);
236 }
237
238 public ArrayList<String> calcularAnt(ElementosLR1 elemento){//
    Produccion produccion, int posicion){
239     ArrayList<String> ant = new ArrayList();
240     ArrayList<Simbolo> consec = elemento.getProduc().getConsec();
241     ArrayList<NoTerminal> nterm = this.gramatica.getNoTerm();
242     int posicion = elemento.getPosicion();
243     if(posicion < consec.size()-1){
244         Simbolo simbolo = consec.get(posicion+1);
245         if(this.gramatica.isTerminal(simbolo.getNombre())){
246             ant.add(simbolo.getNombre());
247             return ant;
248         }else{
249             int j=0;
250             while(j < nterm.size()){
251                 if(nterm.get(j).getNombre().equals(simbolo.
                    getNombre())){
252                     int k=0;
253                     while(k < nterm.get(j).getPrimeros().size()){
254                         ant.add(nterm.get(j).getPrimeros().get(k).
                            getNombre());
255                         k++;

```

```

256         }
257         break;
258     }else{
259         j++;
260     }
261 }
262 return ant;
263 }
264 }else
265     return null;
266 }
267
268 public void agrupamos(ArrayList<ElementosLR1> elementos){
269     int i=1;
270     ArrayList<ElementosLR1> elementosLR = new ArrayList();
271     ElementosLR1 iguales = new ElementosLR1();
272     elementosLR.add(elementos.get(0));
273     while(i < elementos.size()){
274         ElementosLR1 el = elementos.get(i);
275         int j=0;
276         while(j < elementosLR.size()){
277             iguales = null;
278             if(elementosLR.get(j).getPosicion() == el.getPosicion()
279                 && el.getProduc().getConsec().size() ==
280                 elementosLR.get(j).getProduc().getConsec().size()){
281                 int x=0;
282                 int ig=0;
283                 while(x < el.getProduc().getConsec().size()){
284                     if(!elementosLR.get(j).getProduc().getConsec().
285                         get(x).getNombre().equals(el.getProduc().
286                             getConsec().get(x).getNombre())){
287                         break;
288                     }else{
289                         x++;
290                         ig=1;
291                     }
292                 }
293                 if(ig==1)
294                     iguales = elementosLR.get(j);
295             }
296             if(iguales == null)
297                 j++;
298             else
299                 break;
300         }
301         if(iguales == null){
302             elementosLR.add(el);
303         }else{
304             ArrayList<String> ant = new ArrayList();
305             int k=0;
306             while(k < el.getAnticipacion().size()){
307                 ant.add(el.getAnticipacion().get(k));

```

```

305         k++;
306     }
307     int l=0;
308     while(l < iguales.getAnticipacion().size()){
309         ant.add(iguales.getAnticipacion().get(l));
310         l++;
311     }
312     int m=1;
313     ArrayList<String> ant2 = new ArrayList();
314     ant2.add(ant.get(0));
315     while(m < ant.size()){
316         int n=0;
317         int enc = 0;
318         while(n < ant2.size()){
319             if(ant.get(m).equals(ant.get(n))){
320                 enc = 1;
321                 break;
322             }else
323                 n++;
324         }
325         if(enc == 0)
326             ant2.add(ant.get(m));
327         m++;
328     }
329     elementosLR.set(j, new ElementosLR1(el.getProduc(), el.
        getPosicion(), el.getPivote(), ant2));
330     }
331     i++;
332 }
333 this.setElementosLR1(elementosLR);
334 }
335
336 public ArrayList<ElementosLR1> getElementosLR1() {
337     return elementosLR1;
338 }
339
340 public void setElementosLR1(ArrayList<ElementosLR1> elementosLR1) {
341     this.elementosLR1 = elementosLR1;
342 }
343
344 public int getI() {
345     return i;
346 }
347
348 public void setI(int i) {
349     this.i = i;
350 }
351
352 public int getV() {
353     return v;
354 }
355
356 public void setV(int v) {

```

```

357         this.v = v;
358     }
359
360     public int getY() {
361         return y;
362     }
363
364     public void setY(int y) {
365         this.y = y;
366     }
367 }

```

### 2.4.7. Editor.java

```

1  //SimAS / Editor
2  //Editor
3
4  package es.uco.simas.editor;
5
6  import javax.swing.JOptionPane;
7  import es.uco.simas.util.gramatica.*;
8  import java.util.logging.Level;
9  import java.util.logging.Logger;
10 import java.util.ArrayList;
11 import com.itextpdf.text.DocumentException;
12 import es.uco.simas.centroayuda.AcercaDe;
13 import es.uco.simas.simulador.VentanaSimuladorAsc;
14 import es.uco.simas.simulador.VentanaSimuladorDesc;
15 import java.io.File;
16 import javax.swing.DefaultListModel;
17 import javax.swing.JFileChooser;
18 import javax.swing.filechooser.FileNameExtensionFilter;
19
20 /**
21  * @author vanesa
22  */
23 public class Editor extends javax.swing.JFrame {
24     Gramatica gramatica = crearGramatica();
25     Editor ancestor ;
26
27     public Editor() {
28         initComponents();
29
30         this.jMenuItemAsc.setEnabled(false);
31         this.jMenuItemDesc.setEnabled(false);
32         this.jPanel2.setVisible(false);
33         this.jButtonEditar.setEnabled(false);
34         this.jButtonEliminar.setEnabled(false);
35         this.jButtonGuardar.setEnabled(false);
36         this.jButtonSimDesc.setEnabled(false);

```



```

37     this.jMenuItemCerrar.setEnabled(false);
38     this.jMenuItemEditar.setEnabled(false);
39     this.jMenuItemGuardar.setEnabled(false);
40     this.jButtonValidar.setEnabled(false);
41     this.jMenuItemValidar.setEnabled(false);
42     this.jButtonpdf.setEnabled(false);
43     this.jLabelEstado.setEnabled(false);
44     this.jButton1SimAsc.setEnabled(false);
45     this.jButtonSimDesc.setEnabled(false);
46     this.jMenuItemAsc.setEnabled(false);
47     this.jMenuItemDesc.setEnabled(false);
48 }
49
50 public Editor(Gramatica gr) {
51
52     this.gramatica = gr;
53     DefaultListModel model = this.gramatica.getProducciones();
54     DefaultListModel model2 = new DefaultListModel();
55     initComponents();
56     DefaultListModel noTerminales = new DefaultListModel();
57     ArrayList<NoTerminal> noTerm = this.gramatica.getNoTerm();
58
59     this.jButton1SimAsc.setEnabled(false);
60     this.jButtonSimDesc.setEnabled(false);
61     this.jMenuItemAsc.setEnabled(false);
62     this.jMenuItemDesc.setEnabled(false);
63     this.jButtonEditar.setEnabled(true);
64     this.jButtonEliminar.setEnabled(true);
65     this.jButtonGuardar.setEnabled(true);
66     this.jMenuItemCerrar.setEnabled(true);
67     this.jMenuItemEditar.setEnabled(true);
68     this.jMenuItemGuardar.setEnabled(true);
69     this.jButtonValidar.setEnabled(true);
70     this.jMenuItemValidar.setEnabled(true);
71     if (gr.getEstado() == 1) {
72         this.jLabelEstado.setIcon(new javax.swing.ImageIcon(getClass()
73             .getResource("/es/uco/simas/resources/aceptar.png")));
74         this.jLabelEstado.setText("La gramática está validada");
75         this.jButton1SimAsc.setEnabled(true);
76         this.jButtonSimDesc.setEnabled(true);
77         this.jMenuItemAsc.setEnabled(true);
78         this.jMenuItemDesc.setEnabled(true);
79     } else {
80         this.jLabelEstado.setIcon(new javax.swing.ImageIcon(getClass()
81             .getResource("/es/uco/simas/resources/cancelar.png")));
82         this.jLabelEstado.setText("La gramática no está validada");
83         this.jButton1SimAsc.setEnabled(false);
84         this.jButtonSimDesc.setEnabled(false);
85     }
86
87     this.jTextFieldNombre.setText(this.gramatica.getNombre());

```

```

86      this.jTextAreaDesc.setText(this.gramatica.getDescripcion());
87      this.jTextAreaDesc.setLineWrap(true);
88      this.jTextAreaDesc.setWrapStyleWord(true);
89      if(this.gramatica.getSimbInicial()!=null)
90          this.jTextFieldSI.setText(this.gramatica.getSimbInicial());
91      if(this.gramatica.getNoTerminales()!= null){
92          int v=0;
93          while(v < noTerm.size()){
94              noTerminales.addElement(noTerm.get(v).getNombre());
95              v++;
96          }
97          this.jListNT.setModel(noTerminales);
98      }
99      if(this.gramatica.getTerminales() != null)
100          this.jListT.setModel(this.gramatica.getTerminales());
101      if(this.gramatica.getProducciones() != null){
102          int i =0;
103          Object obj;
104          obj = "P {"";
105          model2.addElement(obj);
106          while (i < model.size()){
107              obj = "      "+(i+1)+"      "+model.elementAt(i);
108              model2.addElement(obj);
109              i++;
110          }
111          obj ="}";
112          model2.addElement(obj);
113          this.jListProd.setModel(model2);
114      }
115  }
116
117  /**
118   * This method is called from within the constructor to initialize
119   * the form.
120   * WARNING: Do NOT modify this code. The content of this method is
121   * always
122   * regenerated by the Form Editor.
123   */
124  @SuppressWarnings("unchecked")
125  // <editor-fold defaultstate="collapsed" desc="Generated Code">
126  GEN-BEGIN: initComponents
127  private void initComponents() {
128
129      jPanel1 = new javax.swing.JPanel();
130      jToolBar1 = new javax.swing.JToolBar();
131      jButtonAnadir = new javax.swing.JButton();
132      jButtonAbrir = new javax.swing.JButton();
133      jButtonGuardar = new javax.swing.JButton();
134      jButtonEditar = new javax.swing.JButton();
135      jButtonEliminar = new javax.swing.JButton();
136      jButtonValidar = new javax.swing.JButton();
137      jButtonpdf = new javax.swing.JButton();
138      jSeparator1 = new javax.swing.JToolBar.Separator();

```

```

136      jButtonSimDesc = new javax.swing.JButton();
137      jButton1SimAsc = new javax.swing.JButton();
138      jSeparator2 = new javax.swing.JToolBar.Separator();
139      jButton2 = new javax.swing.JButton();
140      jPanel2 = new javax.swing.JPanel();
141      jLabelNom = new javax.swing.JLabel();
142      jLabelDesc = new javax.swing.JLabel();
143      jLabel1 = new javax.swing.JLabel();
144      jScrollPane1 = new javax.swing.JScrollPane();
145      jListNT = new javax.swing.JList();
146      jLabel2 = new javax.swing.JLabel();
147      jScrollPane2 = new javax.swing.JScrollPane();
148      jListT = new javax.swing.JList();
149      jLabel3 = new javax.swing.JLabel();
150      jLabel4 = new javax.swing.JLabel();
151      jScrollPane3 = new javax.swing.JScrollPane();
152      jListProd = new javax.swing.JList();
153      jTextFieldNombre = new javax.swing.JTextField();
154      jTextFieldSI = new javax.swing.JTextField();
155      jScrollPane4 = new javax.swing.JScrollPane();
156      jTextAreaDesc = new javax.swing.JTextArea();
157      jLabelEstado = new javax.swing.JLabel();
158      jMenuBar1 = new javax.swing.JMenuBar();
159      jMenuItemEditor = new javax.swing.JMenuItem();
160      jMenuItemNuevo = new javax.swing.JMenuItem();
161      jMenuItemAbrir = new javax.swing.JMenuItem();
162      jMenuItemGuardar = new javax.swing.JMenuItem();
163      jMenuItemEditar = new javax.swing.JMenuItem();
164      jMenuItemCerrar = new javax.swing.JMenuItem();
165      jMenuItemValidar = new javax.swing.JMenuItem();
166      jMenuItemSalir = new javax.swing.JMenuItem();
167      jMenuItemSimulador = new javax.swing.JMenuItem();
168      jMenuItemDesc = new javax.swing.JMenuItem();
169      jMenuItemAsc = new javax.swing.JMenuItem();
170      jMenuItemAyuda = new javax.swing.JMenuItem();
171      jMenuItem1 = new javax.swing.JMenuItem();
172      jMenuItem2 = new javax.swing.JMenuItem();
173
174      setDefaultCloseOperation(javax.swing.WindowConstants.
        DISPOSE_ON_CLOSE);
175      setTitle("Editor Gramática Contexto Libre");
176      setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
177
178      jPanel1.setBackground(new java.awt.Color(233, 244, 244));
179
180      jToolBar1.setRollover(true);
181
182      jButtonAnadir.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/nueva.png"))); //
        NOI18N
183      jButtonAnadir.setToolTipText("Nueva Gramática");
184      jButtonAnadir.setFocusable(false);

```

```

185     jButtonAnadir.setHorizontalTextPosition(javax.swing.
        SwingConstants.CENTER);
186     jButtonAnadir.setVerticalTextPosition(javax.swing.
        SwingConstants.BOTTOM);
187     jButtonAnadir.addActionListener(new java.awt.event.
        ActionListener() {
188         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
189             jButtonAnadirActionPerformed(evt);
190         }
191     });
192     jToolBar1.add(jButtonAnadir);
193
194     jButtonAbrir.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/abrir.png"))); //
        NOI18N
195     jButtonAbrir.setToolTipText("Abrir Gram tica");
196     jButtonAbrir.setFocusable(false);
197     jButtonAbrir.setHorizontalTextPosition(javax.swing.
        SwingConstants.CENTER);
198     jButtonAbrir.setVerticalTextPosition(javax.swing.SwingConstants
        .BOTTOM);
199     jButtonAbrir.addActionListener(new java.awt.event.
        ActionListener() {
200         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
201             jButtonAbrirActionPerformed(evt);
202         }
203     });
204     jToolBar1.add(jButtonAbrir);
205
206     jButtonGuardar.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/guardar.png"))); //
        NOI18N
207     jButtonGuardar.setToolTipText("Guardar Gram tica");
208     jButtonGuardar.setFocusable(false);
209     jButtonGuardar.setHorizontalTextPosition(javax.swing.
        SwingConstants.CENTER);
210     jButtonGuardar.setVerticalTextPosition(javax.swing.
        SwingConstants.BOTTOM);
211     jButtonGuardar.addActionListener(new java.awt.event.
        ActionListener() {
212         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
213             jButtonGuardarActionPerformed(evt);
214         }
215     });
216     jToolBar1.add(jButtonGuardar);
217
218     jButtonEditar.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/editar.png"))); //
        NOI18N
219     jButtonEditar.setToolTipText("Editar Gram tica");

```

```

220     jButtonEditar.setFocusable(false);
221     jButtonEditar.setHorizontalTextPosition(javax.swing.
        SwingConstants.CENTER);
222     jButtonEditar.setVerticalTextPosition(javax.swing.
        SwingConstants.BOTTOM);
223     jButtonEditar.addActionListener(new java.awt.event.
        ActionListener() {
224         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
225             jButtonEditarActionPerformed(evt);
226         }
227     });
228     jToolBar1.add(jButtonEditar);
229
230     jButtonEliminar.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/eliminar.png"))); //
        NOI18N
231     jButtonEliminar.setToolTipText("Cerrar Gramática");
232     jButtonEliminar.setFocusable(false);
233     jButtonEliminar.setHorizontalTextPosition(javax.swing.
        SwingConstants.CENTER);
234     jButtonEliminar.setVerticalTextPosition(javax.swing.
        SwingConstants.BOTTOM);
235     jButtonEliminar.addActionListener(new java.awt.event.
        ActionListener() {
236         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
237             jButtonEliminarActionPerformed(evt);
238         }
239     });
240     jToolBar1.add(jButtonEliminar);
241
242     jButtonValidar.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/validar.png"))); //
        NOI18N
243     jButtonValidar.setToolTipText("Validar Gramatica");
244     jButtonValidar.setFocusable(false);
245     jButtonValidar.setHorizontalTextPosition(javax.swing.
        SwingConstants.CENTER);
246     jButtonValidar.setVerticalTextPosition(javax.swing.
        SwingConstants.BOTTOM);
247     jButtonValidar.addActionListener(new java.awt.event.
        ActionListener() {
248         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
249             jMenuItemValidarActionPerformed(evt);
250         }
251     });
252     jToolBar1.add(jButtonValidar);
253
254     jButtonpdf.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/informeGr.png"))); //
        NOI18N

```

```

255     jButtonpdf.setToolTipText("Informe Gramática");
256     jButtonpdf.setFocusable(false);
257     jButtonpdf.setHorizontalTextPosition(javax.swing.SwingConstants
        .CENTER);
258     jButtonpdf.setVerticalTextPosition(javax.swing.SwingConstants.
        BOTTOM);
259     jButtonpdf.addActionListener(new java.awt.event.ActionListener
        () {
260         public void actionPerformed(java.awt.event.ActionEvent evt)
        {
261             jButtonpdfActionPerformed(evt);
262         }
263     });
264     jToolBar1.add(jButtonpdf);
265     jToolBar1.add(jSeparator1);
266
267     jButtonSimDesc.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/similarDesc.png")));
        // NOI18N
268     jButtonSimDesc.setMnemonic('S');
269     jButtonSimDesc.setToolTipText("Simulacion Descendente");
270     jButtonSimDesc.setFocusable(false);
271     jButtonSimDesc.setHorizontalTextPosition(javax.swing.
        SwingConstants.CENTER);
272     jButtonSimDesc.setVerticalTextPosition(javax.swing.
        SwingConstants.BOTTOM);
273     jButtonSimDesc.addActionListener(new java.awt.event.
        ActionListener() {
274         public void actionPerformed(java.awt.event.ActionEvent evt)
        {
275             jButtonSimDescActionPerformed(evt);
276         }
277     });
278     jToolBar1.add(jButtonSimDesc);
279
280     jButton1SimAsc.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/similarAsc.png"))); //
        NOI18N
281     jButton1SimAsc.setMnemonic('S');
282     jButton1SimAsc.setToolTipText("Simulacion Ascendente");
283     jButton1SimAsc.setFocusable(false);
284     jButton1SimAsc.setHorizontalTextPosition(javax.swing.
        SwingConstants.CENTER);
285     jButton1SimAsc.setVerticalTextPosition(javax.swing.
        SwingConstants.BOTTOM);
286     jButton1SimAsc.addActionListener(new java.awt.event.
        ActionListener() {
287         public void actionPerformed(java.awt.event.ActionEvent evt)
        {
288             jButton1SimAscActionPerformed(evt);
289         }
290     });
291     jToolBar1.add(jButton1SimAsc);

```

```

292     jToolBar1.add(jSeparator2);
293
294     jButton2.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/salir.png"))); //
        NOI18N
295     jButton2.setToolTipText("Salir");
296     jButton2.setFocusable(false);
297     jButton2.setHorizontalTextPosition(javax.swing.SwingConstants.
        CENTER);
298     jButton2.setVerticalTextPosition(javax.swing.SwingConstants.
        BOTTOM);
299     jButton2.addActionListener(new java.awt.event.ActionListener()
        {
300         public void actionPerformed(java.awt.event.ActionEvent evt)
        {
301             jButton2ActionPerformed(evt);
302         }
303     });
304     jToolBar1.add(jButton2);
305
306     jPanel2.setBackground(new java.awt.Color(233, 244, 244));
307     jPanel2.setOpaque(false);
308
309     jLabelNom.setFont(new java.awt.Font("Ubuntu", 1, 15)); //
        NOI18N
310     jLabelNom.setForeground(new java.awt.Color(33, 77, 72));
311     jLabelNom.setText("Nombre: ");
312
313     jLabelDesc.setFont(new java.awt.Font("Ubuntu", 1, 15)); //
        NOI18N
314     jLabelDesc.setForeground(new java.awt.Color(33, 77, 72));
315     jLabelDesc.setText("Descripcion: ");
316
317     jLabel1.setFont(new java.awt.Font("Ubuntu", 1, 15)); // NOI18N
318     jLabel1.setForeground(new java.awt.Color(33, 77, 72));
319     jLabel1.setText("Símbolos No Terminales:");
320
321     jScrollPane1.setViewportView(jListNT);
322
323     jLabel2.setFont(new java.awt.Font("Ubuntu", 1, 15)); // NOI18N
324     jLabel2.setForeground(new java.awt.Color(33, 77, 72));
325     jLabel2.setText("Símbolos Terminales:");
326
327     jScrollPane2.setViewportView(jListT);
328
329     jLabel3.setFont(new java.awt.Font("Ubuntu", 1, 15)); // NOI18N
330     jLabel3.setForeground(new java.awt.Color(33, 77, 72));
331     jLabel3.setText("Símbolo inicial: ");
332
333     jLabel4.setFont(new java.awt.Font("Ubuntu", 1, 15)); // NOI18N
334     jLabel4.setForeground(new java.awt.Color(33, 77, 72));
335     jLabel4.setText("Producciones:");
336

```







```

371         .addComponent(jLabelNom)
372         .addGap(34, 34, 34)
373         .addComponent(jTextFieldNombre,
            javax.swing.GroupLayout.
                PREFERRED_SIZE, 258, javax.
                swing.GroupLayout.
                PREFERRED_SIZE)
374         .addGap(50, 50, 50)
375         .addComponent(jLabelEstado))
376     .addComponent(jLabelDesc)
377     .addComponent(jLabel4)
378     .addComponent(jScrollPane3, javax.
        swing.GroupLayout.
            PREFERRED_SIZE, 670, javax.
            swing.GroupLayout.
            PREFERRED_SIZE)
379     .addGroup(jPanel2Layout.
        createSequentialGroup()
            .addComponent(jLabel3)
            .addPreferredGap(javax.swing.
                LayoutStyle.
                ComponentPlacement.RELATED)
            .addComponent(jTextFieldSI,
                javax.swing.GroupLayout.
                    PREFERRED_SIZE, 137, javax.
                    swing.GroupLayout.
                    PREFERRED_SIZE)))
383         .addGap(0, 0, Short.MAX_VALUE)))
384     .addContainerGap(javax.swing.GroupLayout.
        DEFAULT_SIZE, Short.MAX_VALUE))
385     .addGroup(jPanel2Layout.createSequentialGroup()
386         .addComponent(jLabel1)
387         .addPreferredGap(javax.swing.GroupLayout.
            LayoutStyle.
                ComponentPlacement.RELATED, javax.swing.
                GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jLabel2)
            .addGap(0, 0, Short.MAX_VALUE))))
388
389 );
390
391 jPanel2Layout.setVerticalGroup(
392     jPanel2Layout.createParallelGroup(
393         Alignment.LEADING)
394         .addGroup(jPanel2Layout.createSequentialGroup()
395             .addContainerGap()
396             .addGroup(jPanel2Layout.createParallelGroup(
397                 javax.swing.
                    GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabelNom)
                    .addComponent(jTextFieldNombre, javax.swing.
                        GroupLayout.PREFERRED_SIZE, javax.swing.
                        GroupLayout.DEFAULT_SIZE, javax.swing.
                        GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabelEstado))
                .addPreferredGap(javax.swing.GroupLayout.
                    ComponentPlacement.RELATED)

```

```

400         .addComponent(jLabelDesc)
401         .addGap(4, 4, 4)
402         .addComponent(jScrollPane4, javax.swing.GroupLayout.
            PREFERRED_SIZE, 77, javax.swing.GroupLayout.
            PREFERRED_SIZE)
403         .addPreferredGap(javax.swing.LayoutStyle.
            ComponentPlacement.UNRELATED)
404         .addGroup(jPanel2Layout.createParallelGroup(javax.swing
            .GroupLayout.Alignment.BASELINE)
405             .addComponent(jLabel1)
406             .addComponent(jLabel2))
407         .addPreferredGap(javax.swing.LayoutStyle.
            ComponentPlacement.UNRELATED)
408         .addGroup(jPanel2Layout.createParallelGroup(javax.swing
            .GroupLayout.Alignment.LEADING)
409             .addGroup(jPanel2Layout.createSequentialGroup())
410                 .addComponent(jScrollPane1, javax.swing.
                    GroupLayout.PREFERRED_SIZE, 144, javax.
                    swing.GroupLayout.PREFERRED_SIZE)
411                 .addGap(0, 0, Short.MAX_VALUE))
412             .addComponent(jScrollPane2))
413         .addGap(12, 12, 12)
414         .addGroup(jPanel2Layout.createParallelGroup(javax.swing
            .GroupLayout.Alignment.BASELINE)
415             .addComponent(jLabel3)
416             .addComponent(jTextFieldSI, javax.swing.GroupLayout
                .PREFERRED_SIZE, javax.swing.GroupLayout.
                DEFAULT_SIZE, javax.swing.GroupLayout.
                PREFERRED_SIZE))
417         .addPreferredGap(javax.swing.LayoutStyle.
            ComponentPlacement.RELATED)
418         .addComponent(jLabel4)
419         .addPreferredGap(javax.swing.LayoutStyle.
            ComponentPlacement.RELATED)
420         .addComponent(jScrollPane3, javax.swing.GroupLayout.
            PREFERRED_SIZE, 154, javax.swing.GroupLayout.
            PREFERRED_SIZE)
421         .addContainerGap())
422     );
423
424     javax.swing.GroupLayout jPanel1Layout = new javax.swing.
        GroupLayout(jPanel1);
425     jPanel1.setLayout(jPanel1Layout);
426     jPanel1Layout.setHorizontalGroup(
427         jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
            Alignment.LEADING)
428             .addComponent(jToolBar1, javax.swing.GroupLayout.
                DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
                Short.MAX_VALUE)
429             .addGroup(jPanel1Layout.createSequentialGroup())
430                 .addComponent(jPanel2, javax.swing.GroupLayout.
                    DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
                    Short.MAX_VALUE)

```

```

431         .addContainerGap() )
432     );
433     jPanel1Layout.setVerticalGroup(
434         jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
            Alignment.LEADING)
435         .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            jPanel1Layout.createSequentialGroup()
436             .addComponent(jToolBar1, javax.swing.GroupLayout.
                PREFERRED_SIZE, 35, javax.swing.GroupLayout.
                PREFERRED_SIZE)
437             .addPreferredGap(javax.swing.LayoutStyle.
                ComponentPlacement.RELATED)
438             .addComponent(jPanel2, javax.swing.GroupLayout.
                PREFERRED_SIZE, javax.swing.GroupLayout.
                DEFAULT_SIZE, javax.swing.GroupLayout.
                PREFERRED_SIZE)
439             .addContainerGap(184, Short.MAX_VALUE))
440     );
441
442     jMenuItemEditor.setMnemonic('E');
443     jMenuItemEditor.setText(" Editor");
444
445     jMenuItemNuevo.setMnemonic('N');
446     jMenuItemNuevo.setText(" Nueva Gramática");
447     jMenuItemNuevo.addActionListener(new java.awt.event.
        ActionListener() {
448         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
449             jMenuItemNuevoActionPerformed(evt);
450             jButtonAnadirActionPerformed(evt);
451         }
452     });
453     jMenuItemEditor.add(jMenuItemNuevo);
454
455     jMenuItemAbrir.setMnemonic('A');
456     jMenuItemAbrir.setText(" Abrir Gramática");
457     jMenuItemAbrir.addActionListener(new java.awt.event.
        ActionListener() {
458         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
459             jButtonAbrirActionPerformed(evt);
460         }
461     });
462     jMenuItemEditor.add(jMenuItemAbrir);
463
464     jMenuItemGuardar.setText(" Guardar Gramática");
465     jMenuItemGuardar.addActionListener(new java.awt.event.
        ActionListener() {
466         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
467             jButtonGuardarActionPerformed(evt);
468         }
469     });

```

```

470     jMenuItemGuardar.add(jMenuItemGuardar);
471
472     jMenuItemEditar.setMnemonic('D');
473     jMenuItemEditar.setText(" Editar  GramÃ¡tica");
474     jMenuItemEditar.addActionListener(new java.awt.event.
        ActionListener() {
475         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
476             jButtonEditarActionPerformed(evt);
477         }
478     });
479     jMenuItemEditor.add(jMenuItemEditar);
480
481     jMenuItemCerrar.setMnemonic('C');
482     jMenuItemCerrar.setText(" Cerrar  GramÃ¡tica");
483     jMenuItemCerrar.addActionListener(new java.awt.event.
        ActionListener() {
484         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
485             jButtonEliminarActionPerformed(evt);
486         }
487     });
488     jMenuItemEditor.add(jMenuItemCerrar);
489
490     jMenuItemValidar.setMnemonic('V');
491     jMenuItemValidar.setText(" Validar  GramÃ¡tica");
492     jMenuItemValidar.addActionListener(new java.awt.event.
        ActionListener() {
493         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
494             jMenuItemValidarActionPerformed(evt);
495         }
496     });
497     jMenuItemEditor.add(jMenuItemValidar);
498
499     jMenuItemSalir.setText(" Salir");
500     jMenuItemSalir.addActionListener(new java.awt.event.
        ActionListener() {
501         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
502             jButton2ActionPerformed(evt);
503         }
504     });
505     jMenuItemEditor.add(jMenuItemSalir);
506
507     jMenuBar1.add(jMenuItemEditor);
508
509     jMenuItemSimulador.setMnemonic('S');
510     jMenuItemSimulador.setText(" Simulador");
511
512     jMenuItemDesc.setText(" Simulacion  Descendente");
513     jMenuItemDesc.addActionListener(new java.awt.event.
        ActionListener() {

```

```

514         public void actionPerformed(java.awt.event.ActionEvent evt)
515         {
516             jButtonSimDescActionPerformed(evt);
517         }
518     });
519     jMenuSimulador.add(jMenuItemDesc);
520     jMenuItemAsc.setText("Simulacion Ascendente");
521     jMenuItemAsc.addActionListener(new java.awt.event.
522         ActionListener() {
523             public void actionPerformed(java.awt.event.ActionEvent evt)
524             {
525                 jButton1SimAscActionPerformed(evt);
526             }
527         });
528     jMenuSimulador.add(jMenuItemAsc);
529     jMenuBar1.add(jMenuSimulador);
530     jMenuAyuda.setText("Ayuda");
531     jMenuItem1.setText("Centro de Ayuda");
532     jMenuItem1.addActionListener(new java.awt.event.ActionListener
533         () {
534             public void actionPerformed(java.awt.event.ActionEvent evt)
535             {
536                 jMenuItem1ActionPerformed(evt);
537             }
538         });
539     jMenuAyuda.add(jMenuItem1);
540     jMenuItem2.setText("Acerca de ...");
541     jMenuItem2.addActionListener(new java.awt.event.ActionListener
542         () {
543             public void actionPerformed(java.awt.event.ActionEvent evt)
544             {
545                 jMenuItem2ActionPerformed(evt);
546             }
547         });
548     jMenuAyuda.add(jMenuItem2);
549     jMenuBar1.add(jMenuAyuda);
550     setJMenuBar(jMenuBar1);
551     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
552         getContentPane());
553     getContentPane().setLayout(layout);
554     layout.setHorizontalGroup(
555         layout.createParallelGroup(javax.swing.GroupLayout.
556             Alignment.LEADING)
557         .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE
558             , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE

```

```

557         );
558     layout.setVerticalGroup(
559         layout.createParallelGroup(javax.swing.GroupLayout.
            Alignment.LEADING)
560         .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE
            , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE
            )
561     );
562
563     pack();
564 }// </editor-fold> //GEN-END: initComponents
565
566 private void jMenuItemNuevoActionPerformed(java.awt.event.
    ActionEvent evt) { //GEN-FIRST:
        event_jMenuItemNuevoActionPerformed
567         // TODO add your handling code here:
568     } //GEN-LAST: event_jMenuItemNuevoActionPerformed
569
570 private void jButtonAnadirActionPerformed(java.awt.event.
    ActionEvent evt) { //GEN-FIRST:
        event_jButtonAnadirActionPerformed
571         VentanaCreacionGramatica p1 = new VentanaCreacionGramatica(this
            , null);
572         p1.setVisible(true);
573         p1.setLocationRelativeTo(null);
574     } //GEN-LAST: event_jButtonAnadirActionPerformed
575
576 private void jButtonEditarActionPerformed(java.awt.event.
    ActionEvent evt) { //GEN-FIRST:
        event_jButtonEditarActionPerformed
577
578         VentanaCreacionGramatica p1 = new VentanaCreacionGramatica(this
            , this.gramatica);
579         p1.setVisible(true);
580         p1.setLocationRelativeTo(null);
581     } //GEN-LAST: event_jButtonEditarActionPerformed
582
583 private void jButtonSimDescActionPerformed(java.awt.event.
    ActionEvent evt) { //GEN-FIRST:
        event_jButtonSimDescActionPerformed
584         Gramatica gr2 = new Gramatica();
585         gr2 = this.gramatica;
586         VentanaSimuladorDesc simdesc = new VentanaSimuladorDesc(gr2) ;
587         simdesc.setVisible(true);
588         simdesc.setLocationRelativeTo(null);
589     } //GEN-LAST: event_jButtonSimDescActionPerformed
590
591 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt
    ) { //GEN-FIRST: event_jButton2ActionPerformed
592         int conf = JOptionPane.showConfirmDialog(null, "¿Desea salir
            de SimAS?", "Salir", JOptionPane.YES_NO_OPTION);
593

```

```

594         if ( conf==0)
595             System.exit(0);
596     }
597
598     private void cerrarVentana(java.awt.event.WindowEvent evt) {
599         int conf = JOptionPane.showConfirmDialog(null, "¿Desea salir
        de SimAS?", "Salir",JOptionPane.YES_NO_OPTION);
600
601         if ( conf==0)
602             this.dispose();
603     } //GEN-LAST: event_jButtonon2ActionPerformed
604
605     private void jButtonGuardarActionPerformed(java.awt.event.
        ActionEvent evt) { //GEN-FIRST:
        event_jButtononGuardarActionPerformed
606         this.grabarGramatica();
607     } //GEN-LAST: event_jButtononGuardarActionPerformed
608
609     private void jButtonAbrirActionPerformed(java.awt.event.ActionEvent
        evt) { //GEN-FIRST: event_jButtononAbrirActionPerformed
610         this.cargarGramatica();
611     } //GEN-LAST: event_jButtononAbrirActionPerformed
612
613     private void jButtonEliminarActionPerformed(java.awt.event.
        ActionEvent evt) { //GEN-FIRST:
        event_jButtononEliminarActionPerformed
614         int conf = JOptionPane.showConfirmDialog(null, "¿Desea cerrar
        la gramática "+this.gramatica.getNombre()+"?", "Cerrar
        Gramática",JOptionPane.YES_NO_OPTION);
615
616         if ( conf==0){
617             JOptionPane.showConfirmDialog(null, "Gramática cerrada", "
        Cerrar Gramática", JOptionPane.DEFAULT_OPTION);
618             this.dispose();
619             this.gramatica.removeAll();
620             Editor editor = new Editor();
621             editor.setVisible(true);
622             editor.setLocationRelativeTo(null);
623         }
624
625     } //GEN-LAST: event_jButtononEliminarActionPerformed
626
627     private void jMenuItemValidarActionPerformed(java.awt.event.
        ActionEvent evt) { //GEN-FIRST:
        event_jMenuItemValidarActionPerformed
628         if (this.gramatica != null)
629             this.validarGramatica(this.gramatica);
630     } //GEN-LAST: event_jMenuItemValidarActionPerformed
631
632     private void jButtonpdfActionPerformed(java.awt.event.ActionEvent
        evt) { //GEN-FIRST: event_jButtononpdfActionPerformed
633

```

```

634     Boolean resultado= null;
635     FileNameExtensionFilter filtro= null;
636     JFileChooser selector= null;
637
638     JFileChooser chooser = new JFileChooser();
639     selector=chooser;
640     FileNameExtensionFilter extension = new FileNameExtensionFilter
        ("Informes de gramática (.pdf)",new String []
641         {"pdf"});
642     filtro=extension;
643     selector.setFileFilter(filtro);
644     File fichero = new File("informeGramatica.pdf");
645     selector.setSelectedFile(fichero);
646     if(selector.showSaveDialog(null)==0) {
647         try {
648             resultado=this.gramatica.generarInforme(selector.
                getSelectedFile().toString());
649         } catch (DocumentException ex) {
650             Logger.getLogger(Editor.class.getName()).log(Level.
                SEVERE, null, ex);
651         }
652     }
653 }//GEN-LAST:event_jButtononpdfActionPerformed
654
655 private void jButton1SimAscActionPerformed(java.awt.event.
    ActionEvent evt) { //GEN-FIRST:
    event_jButton1SimAscActionPerformed
656     VentanaSimuladorAsc simasc = new VentanaSimuladorAsc(this.
        gramatica) ;
657     simasc.setVisible(true);
658     simasc.setLocationRelativeTo(null);
659 }//GEN-LAST:event_jButton1SimAscActionPerformed
660
661 private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_jMenuItem1ActionPerformed
        // TODO add your handling code here:
662 }//GEN-LAST:event_jMenuItem1ActionPerformed
663
664
665 private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST:event_jMenuItem2ActionPerformed
        AcercaDe acerca = new AcercaDe();
666     acerca.setVisible(true);
667     acerca.setLocationRelativeTo(null);
668 }//GEN-LAST:event_jMenuItem2ActionPerformed
669
670
671 public void cargarGramatica(){
672     Gramatica gr = new Gramatica();
673     gr = this.gramatica.cargarGramatica();
674     if(gr !=null){
675         this.gramatica = gr;
676         this.dispose();
677         Editor edit = new Editor(this.gramatica);
678         edit.setVisible(true);

```



```

679         edit.setLocationRelativeTo(null);
680     }
681 }
682
683 public void grabarGramatica(){
684     int i = this.gramatica.guardarGramatica();
685     if (i==1){
686         JOptionPane.showConfirmDialog(null,"Gramática guardada
            correctamente", "Guardar Gramática",JOptionPane.
            CLOSED_OPTION);
687     }
688 }
689
690 private Gramatica crearGramatica(){
691     return new Gramatica();
692 }
693
694 public Gramatica getGramatica() {
695     return this.gramatica;
696 }
697
698 public void setGramatica(Gramatica gramatica) {
699     this.gramatica = gramatica;
700 }
701
702 public void actualizarVisualizacion(){
703     this.jListNT.setModel(this.gramatica.getNoTerminales());
704     this.jListProd.setModel(this.gramatica.getProducciones());
705     this.jListT.setModel(this.gramatica.getTerminales());
706 }
707
708 public void validarGramatica(Gramatica gramatica) {
709     ArrayList<String> mensajesError = new ArrayList();
710     Editor panelEditor= null;
711     String mensaje= null;
712     int estadoValidacion= 0;
713
714     mensajesError = gramatica.validarGramatica();
715     estadoValidacion= gramatica.getEstado();
716
717     if(estadoValidacion == 1)    {
718         gramatica.setEstado(1);
719         JOptionPane.showConfirmDialog(null , "La gramática está
            validada. ", "Gramática Validada",JOptionPane.
            CLOSED_OPTION);
720         this.dispose();
721         Editor editor = new Editor(gramatica);
722         editor.setVisible(true);
723         editor.setLocationRelativeTo(null);
724     } else {
725         gramatica.setEstado(-1);
726         JOptionPane.showConfirmDialog(null , "<html> <h4>Se han
            detectado los siguientes errores: </h4><br></br>" +

```

```

mensajesError+"</html>", "Error. Gramática No Validada",
JOptionPane.CLOSED_OPTION);
727     }
728
729     }
730     // Variables declaration – do not modify//GEN-BEGIN:variables
731     private javax.swing.JButton jButton1SimAsc;
732     private javax.swing.JButton jButton2;
733     private javax.swing.JButton jButtonAbrir;
734     private javax.swing.JButton jButtonAnadir;
735     private javax.swing.JButton jButtonEditar;
736     private javax.swing.JButton jButtonEliminar;
737     private javax.swing.JButton jButtonGuardar;
738     private javax.swing.JButton jButtonSimDesc;
739     private javax.swing.JButton jButtonValidar;
740     private javax.swing.JButton jButtonpdf;
741     private javax.swing.JLabel jLabel1;
742     private javax.swing.JLabel jLabel2;
743     private javax.swing.JLabel jLabel3;
744     private javax.swing.JLabel jLabel4;
745     private javax.swing.JLabel jLabelDesc;
746     private javax.swing.JLabel jLabelEstado;
747     private javax.swing.JLabel jLabelNom;
748     private javax.swing.JList jListNT;
749     private javax.swing.JList jListProd;
750     private javax.swing.JList jListT;
751     private javax.swing.JMenu jMenuAyuda;
752     private javax.swing.JMenuBar jMenuBar1;
753     private javax.swing.JMenu jMenuEditor;
754     private javax.swing.JMenuItem jMenuItem1;
755     private javax.swing.JMenuItem jMenuItem2;
756     private javax.swing.JMenuItem jMenuItemAbrir;
757     private javax.swing.JMenuItem jMenuItemAsc;
758     private javax.swing.JMenuItem jMenuItemCerrar;
759     private javax.swing.JMenuItem jMenuItemDesc;
760     private javax.swing.JMenuItem jMenuItemEditar;
761     private javax.swing.JMenuItem jMenuItemGuardar;
762     private javax.swing.JMenuItem jMenuItemNuevo;
763     private javax.swing.JMenuItem jMenuItemSalir;
764     private javax.swing.JMenuItem jMenuItemValidar;
765     private javax.swing.JMenu jMenuSimulador;
766     private javax.swing.JPanel jPanel1;
767     private javax.swing.JPanel jPanel2;
768     private javax.swing.JScrollPane jScrollPane1;
769     private javax.swing.JScrollPane jScrollPane2;
770     private javax.swing.JScrollPane jScrollPane3;
771     private javax.swing.JScrollPane jScrollPane4;
772     private javax.swing.JToolBar.Separator jSeparator1;
773     private javax.swing.JToolBar.Separator jSeparator2;
774     private javax.swing.JTextArea jTextAreaDesc;
775     private javax.swing.JTextField jTextFieldNombre;
776     private javax.swing.JTextField jTextFieldSI;
777     private javax.swing.JToolBar jToolBar1;

```

```
778 // End of variables declaration//GEN-END:variables
779 }
```

## 2.4.8. ElementosLALR.java

```
1 //SimAS / editor
2 // ElementosLALR
3
4 package es.uco.simas.editor;
5
6 import es.uco.simas.util.gramatica.Produccion;
7 import java.util.ArrayList;
8
9 /**
10  * @author vanesa
11  */
12 public class ElementosLALR {
13
14     Produccion produc;
15     int posicion;
16     ArrayList<String> anticipacion = new ArrayList();
17     String pivote;
18
19     public ElementosLALR(Produccion produc, int posicion, String pivote
20         , ArrayList<String> anticipacion){
21         this.produc = produc;
22         this.posicion = posicion;
23         this.pivote = pivote;
24         this.anticipacion = anticipacion;
25     }
26     public ElementosLALR(){
27
28     }
29     public Produccion getProduc() {
30         return produc;
31     }
32
33     public void setProduc(Produccion produc) {
34         this.produc = produc;
35     }
36
37     public int getPosicion() {
38         return posicion;
39     }
40
41     public void setPosicion(int posicion) {
42         this.posicion = posicion;
43     }
44 }
```

```
45     public ArrayList<String> getAnticipacion() {
46         return anticipacion;
47     }
48
49     public void setAnticipacion(ArrayList<String> anticipacion) {
50         this.anticipacion = anticipacion;
51     }
52
53     public String getPivote() {
54         return pivote;
55     }
56
57     public void setPivote(String pivote) {
58         this.pivote = pivote;
59     }
60 }
```

### 2.4.9. ElementosLR0.java

```
1  //SimAS / editor
2  // ElementosLR0
3
4  package es.uco.simas.editor;
5
6  import es.uco.simas.util.gramatica.Produccion;
7
8  /**
9   * @author vanesa
10  */
11 public class ElementosLR0 {
12     public Produccion produc;
13     int posicion;
14     String pivote;
15
16     public ElementosLR0(Produccion produc, int posicion, String pivote)
17     {
18         this.produc = produc;
19         this.posicion = posicion;
20         this.pivote = pivote;
21     }
22
23     public Produccion getProduc() {
24         return produc;
25     }
26
27     public void setProduc(Produccion produc) {
28         this.produc = produc;
29     }
30
31     public int getPosicion() {
```

```
31         return posicion;
32     }
33
34     public void setPosicion(int posicion) {
35         this.posicion = posicion;
36     }
37
38     public String getPivote() {
39         return pivote;
40     }
41
42     public void setPivote(String pivote) {
43         this.pivote = pivote;
44     }
45 }
```

#### 2.4.10. ElementosLR1.java

```
1 //SimAS / editor
2 // ElementosLR1
3
4 package es.uco.simas.editor;
5
6 import es.uco.simas.util.gramatica.Produccion;
7 import java.util.ArrayList;
8
9 /**
10  * @author vanesa
11  */
12 public class ElementosLR1 {
13     Produccion produc;
14     int posicion;
15     ArrayList<String> anticipacion = new ArrayList();
16     String pivote;
17
18     public ElementosLR1(Produccion produc, int posicion, String pivote,
19         ArrayList<String> anticipacion){
20         this.produc = produc;
21         this.posicion = posicion;
22         this.pivote = pivote;
23         this.anticipacion = anticipacion;
24     }
25     public ElementosLR1(){
26     }
27
28     public Produccion getProduc() {
29         return produc;
30     }
31 }
```

```

32     public void setProduc(Produccion produc) {
33         this.produc = produc;
34     }
35
36     public int getPosicion() {
37         return posicion;
38     }
39
40     public void setPosicion(int posicion) {
41         this.posicion = posicion;
42     }
43
44     public ArrayList<String> getAnticipacion() {
45         return anticipacion;
46     }
47
48     public void setAnticipacion(ArrayList<String> anticipacion) {
49         this.anticipacion = anticipacion;
50     }
51
52     public String getPivote() {
53         return pivote;
54     }
55
56     public void setPivote(String pivote) {
57         this.pivote = pivote;
58     }
59 }

```

### 2.4.11. FuncionError.java

```

1  // SimAS / Editor
2  //FuncionError
3
4  package es.uco.simas.editor;
5
6  import es.uco.simas.util.gramatica.Terminal;
7  /**
8   * @author vanesa
9   */
10 public class FuncionError {
11
12     public static final int INSERTAR_ENTRADA =1;
13     public static final int BORRAR_ENTRADA =2;
14     public static final int MODIFICAR_ENTRADA =3;
15     public static final int INSERTAR_PILA =4;
16     public static final int BORRAR_PILA =5;
17     public static final int MODIFICAR_PILA =6;
18     public static final int TERMINAR_ANALISIS =7;
19     private int identificador ;

```

```
20     private int accion ;
21     private String mensaje ;
22     private Terminal simbolo ;
23
24     public FuncionError(int id, int acc, String mensaje){
25         this.identificador = id;
26         this.accion = acc;
27         this.mensaje = mensaje;
28     }
29     public FuncionError(){
30
31     }
32
33     public int getIdentificador() {
34         return identificador;
35     }
36
37     public void setIdentificador(int identificador) {
38         this.identificador = identificador;
39     }
40
41     public String getMensaje() {
42         return mensaje;
43     }
44
45     public void setMensaje(String mensaje) {
46         this.mensaje = mensaje;
47     }
48
49     public int getAccion() {
50         return accion;
51     }
52
53     public void setAccion(int accion) {
54         this.accion = accion;
55     }
56
57     public Terminal getSimbolo() {
58         return simbolo;
59     }
60
61     public void setSimbolo(Terminal simbolo) {
62         this.simbolo = simbolo;
63     }
64 }
```

#### 2.4.12. PanelCreacionGramaticaPaso1.java

```
1 // SimAS / Editor
2 // Panel Creacion Gramatica Paso 1
```

```

3
4 package es.uco.simas.editor;
5
6 import javax.swing.JOptionPane;
7
8 /**
9  * @author vanesa
10 */
11 public class PanelCreacionGramaticaPaso1 extends javax.swing.JPanel {
12
13     private final VentanaCreacionGramatica ventanaPadre ;
14
15     public PanelCreacionGramaticaPaso1(VentanaCreacionGramatica
16         ventanaPadre) {
17         initComponents();
18         this.ventanaPadre =ventanaPadre;
19         this.jTextNombre.requestFocus();
20         this.jTextDescripcion.setLineWrap(true);
21     }
22
23     public String getNombreGramatica( ) {
24         return this.jTextNombre.getText();
25     }
26
27     public void setNombre(String nombre){
28         this.jTextNombre.setText(nombre);
29     }
30
31     public String getDescripcionGramatica( ) {
32         return this.jTextDescripcion.getText();
33     }
34
35     public void setDescripcion (String descripcion){
36         this.jTextDescripcion.setText(descripcion);
37     }
38
39     /**
40      * This method is called from within the constructor to initialize
41      * the form.
42      * WARNING: Do NOT modify this code. The content of this method is
43      * always
44      * regenerated by the Form Editor.
45      */
46
47     // <editor-fold defaultstate="collapsed" desc="Generated Code">
48     GEN-BEGIN: initComponents
49     private void initComponents() {
50
51         jLabel4 = new javax.swing.JLabel();
52         jTextNombre = new javax.swing.JTextField();
53         jLabel5 = new javax.swing.JLabel();
54         jScrollPane2 = new javax.swing.JScrollPane();
55         jTextDescripcion = new javax.swing.JTextArea();
56         jButtonSiguiente = new javax.swing.JButton();
57         jButtonAnterior = new javax.swing.JButton();

```



```

52      jButtonCancelar = new javax.swing.JButton();
53      jLabel1 = new javax.swing.JLabel();
54      jButtonUltimo = new javax.swing.JButton();
55      jButton1 = new javax.swing.JButton();
56
57      setBackground(new java.awt.Color(233, 244, 244));
58      setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
59
60      jLabel4.setFont(new java.awt.Font("Ubuntu", 1, 15)); // NOI18N
61      jLabel4.setForeground(new java.awt.Color(33, 77, 72));
62      jLabel4.setText("Nombre de la Gramática:");
63
64      jLabel5.setFont(new java.awt.Font("Ubuntu", 1, 15)); // NOI18N
65      jLabel5.setForeground(new java.awt.Color(33, 77, 72));
66      jLabel5.setText("Descripción:");
67
68      jTextDescripcion.setColumns(20);
69      jTextDescripcion.setRows(5);
70      jScrollPane2.setViewportView(jTextDescripcion);
71
72      jButtonSiguiente.setForeground(new java.awt.Color(33, 77, 72));
73      jButtonSiguiente.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/siguiente.png"))); //
        NOI18N
74      jButtonSiguiente.setToolTipText("Siguiente");
75      jButtonSiguiente.addActionListener(new java.awt.event.
        ActionListener() {
76          public void actionPerformed(java.awt.event.ActionEvent evt)
77          {
78              jButtonSiguienteActionPerformed(evt);
79          }
80      });
81
82      jButtonAnterior.setForeground(new java.awt.Color(33, 77, 72));
83      jButtonAnterior.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/anterior.png"))); //
        NOI18N
84      jButtonAnterior.setEnabled(false);
85
86      jButtonCancelar.setForeground(new java.awt.Color(33, 77, 72));
87      jButtonCancelar.setText("Cancelar");
88      jButtonCancelar.addActionListener(new java.awt.event.
        ActionListener() {
89          public void actionPerformed(java.awt.event.ActionEvent evt)
90          {
91              jButtonCancelarActionPerformed(evt);
92          }
93      });
94
95      jLabel1.setFont(new java.awt.Font("Ubuntu", 1, 18)); // NOI18N
96      jLabel1.setForeground(new java.awt.Color(33, 77, 72));
97      jLabel1.setText("Datos de la Gramática");

```

```

97      jButtonUltimo.setIcon(new javax.swing.ImageIcon(getClass().
           getResource("/es/uco/simas/resources/ultimo.png"))); //
           NOI18N
98      jButtonUltimo.addActionListener(new java.awt.event.
           ActionListener() {
99          public void actionPerformed(java.awt.event.ActionEvent evt)
           {
100              jButtonUltimoActionPerformed(evt);
101          }
102      });
103
104      jButton1.setIcon(new javax.swing.ImageIcon(getClass().
           getResource("/es/uco/simas/resources/primer.png"))); //
           NOI18N
105      jButton1.setToolTipText("");
106      jButton1.setEnabled(false);
107
108      javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
           this);
109      this.setLayout(layout);
110      layout.setHorizontalGroup(
111          layout.createParallelGroup(javax.swing.GroupLayout.
           Alignment.LEADING)
112          .addGroup(layout.createSequentialGroup()
113              .addGap(42, 42, 42)
114              .addGroup(layout.createParallelGroup(javax.swing.
           GroupLayout.Alignment.TRAILING)
115                  .addGroup(layout.createSequentialGroup()
116                      .addComponent(jButtonCancelar)
117                      .addPreferredGap(javax.swing.LayoutStyle.
           ComponentPlacement.RELATED, javax.swing.
           GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
118                      .addComponent(jButton1)
119                      .addPreferredGap(javax.swing.LayoutStyle.
           ComponentPlacement.RELATED)
120                      .addComponent(jButtonAnterior)
121                      .addPreferredGap(javax.swing.LayoutStyle.
           ComponentPlacement.RELATED)
122                      .addComponent(jButtonSiguiente)
123                      .addPreferredGap(javax.swing.LayoutStyle.
           ComponentPlacement.RELATED)
124                      .addComponent(jButtonUltimo))
125                  .addComponent(jScrollPane2, javax.swing.GroupLayout.
           Alignment.LEADING)
126                  .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
           layout.createSequentialGroup()
127                      .addComponent(jLabel4)
128                      .addPreferredGap(javax.swing.LayoutStyle.
           ComponentPlacement.RELATED)
129                      .addComponent(jTextNombre))
130                  .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
           layout.createSequentialGroup()
131                      .addComponent(jLabel5)

```

```

132         .addGap(0, 0, Short.MAX_VALUE)))
133     .addGap(42, 42, 42))
134     .addGroup(layout.createSequentialGroup())
135     .addGap(218, 218, 218)
136     .addComponent(jLabel1)
137     .addContainerGap(218, Short.MAX_VALUE))
138 );
139 layout.setVerticalGroup(
140     layout.createParallelGroup(javax.swing.GroupLayout.
141         Alignment.LEADING)
142     .addGroup(layout.createSequentialGroup())
143     .addGap(10, 10, 10)
144     .addComponent(jLabel1)
145     .addGap(18, 18, 18)
146     .addGroup(layout.createParallelGroup(javax.swing.
147         GroupLayout.Alignment.BASELINE)
148         .addComponent(jLabel4)
149         .addComponent(jTextNombre, javax.swing.GroupLayout.
150             PREFERRED_SIZE, javax.swing.GroupLayout.
151             DEFAULT_SIZE, javax.swing.GroupLayout.
152             PREFERRED_SIZE))
153     .addGroup(layout.createParallelGroup(javax.swing.
154         GroupLayout.Alignment.LEADING)
155         .addGroup(layout.createSequentialGroup())
156         .addGap(25, 25, 25)
157         .addComponent(jLabel5)
158         .addPreferredGap(javax.swing.LayoutStyle.
159             ComponentPlacement.RELATED)
160         .addComponent(jScrollPane2, javax.swing.
161             GroupLayout.PREFERRED_SIZE, 137, javax.
162             swing.GroupLayout.PREFERRED_SIZE)
163         .addPreferredGap(javax.swing.LayoutStyle.
164             ComponentPlacement.RELATED, 58, Short.
165             MAX_VALUE)
166         .addGroup(layout.createParallelGroup(javax.
167             swing.GroupLayout.Alignment.LEADING)
168             .addComponent(jButtonAnterior, javax.swing.
169                 GroupLayout.Alignment.TRAILING)
170             .addComponent(jButtonSiguiente, javax.swing.
171                 GroupLayout.Alignment.TRAILING)
172             .addGroup(layout.createParallelGroup(javax.
173                 swing.GroupLayout.Alignment.BASELINE)
174                 .addComponent(jButtonCancelar)
175                 .addComponent(jButton1))))
176     .addGroup(layout.createSequentialGroup())
177     .addPreferredGap(javax.swing.LayoutStyle.
178         ComponentPlacement.RELATED, javax.swing.
179         GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
180     .addComponent(jButtonUltimo)))
181     .addContainerGap())
182 );
183 } // </editor-fold> // GEN-END: initComponents
184

```

```

168     private void jButtonSiguienteActionPerformed(java.awt.event.
        ActionEvent evt) {//GEN-FIRST:
        event_jButtonSiguienteActionPerformed
169         if(("".equals(jTextNombre.getText())) || ("".equals(
            jTextDescripcion.getText()))){
170             JOptionPane.showConfirmDialog(null, "Los campos Nombre y
                Descripcion no pueden estar vacÃos.", "Error",
                JOptionPane.CLOSED_OPTION);
171         }else{
172             this.ventanaPadre.cambiarPaso(2);
173         }
174     }//GEN-LAST:event_jButtonSiguienteActionPerformed
175
176     private void jButtonCancelarActionPerformed(java.awt.event.
        ActionEvent evt) {//GEN-FIRST:
        event_jButtonCancelarActionPerformed
177         int conf = JOptionPane.showConfirmDialog(null, "Â¿Desea salir
            de la edicion de la gramÃtica?", "Salir",JOptionPane.
            YES_NO_OPTION);
178
179         if(conf==0)
180             this.ventanaPadre.finalizarAsistente();//dispose();
181     }//GEN-LAST:event_jButtonCancelarActionPerformed
182
183     private void jButtonUltimoActionPerformed(java.awt.event.
        ActionEvent evt) {//GEN-FIRST:
        event_jButtonUltimoActionPerformed
184         if(("".equals(jTextNombre.getText())) || ("".equals(
            jTextDescripcion.getText()))){
185             JOptionPane.showConfirmDialog(null, "Los campos Nombre y
                Descripcion no pueden estar vacÃos.", "Error",
                JOptionPane.CLOSED_OPTION);
186         }else{
187             this.ventanaPadre.cambiarPaso(4);
188         }
189     }//GEN-LAST:event_jButtonUltimoActionPerformed
190
191     // Variables declaration - do not modify//GEN-BEGIN:variables
192     private javax.swing.JButton jButton1;
193     private javax.swing.JButton jButtonAnterior;
194     private javax.swing.JButton jButtonCancelar;
195     private javax.swing.JButton jButtonSiguiente;
196     private javax.swing.JButton jButtonUltimo;
197     private javax.swing.JLabel jLabel1;
198     private javax.swing.JLabel jLabel4;
199     private javax.swing.JLabel jLabel5;
200     private javax.swing.JScrollPane jScrollPane2;
201     private javax.swing.JTextArea jTextDescripcion;
202     private javax.swing.JTextField jTextNombre;
203     // End of variables declaration//GEN-END:variables
204 }

```

## 2.4.13. PanelCreacionGramaticaPaso2.java

```

1 //SimAS / Editor
2 // Panel Creacion Gramatica Paso 2
3
4 package es.uco.simas.editor;
5
6 import javax.swing.JOptionPane;
7 import javax.swing.DefaultListModel;
8
9 /**
10  * @author vanesa
11  */
12 public class PanelCreacionGramaticaPaso2 extends javax.swing.JPanel {
13
14     private VentanaCreacionGramatica ventanaPadre ;
15
16     public PanelCreacionGramaticaPaso2(VentanaCreacionGramatica
17         ventanaPadre) {
18         initComponents();
19         this.ventanaPadre =ventanaPadre;
20     }
21     /**
22      * This method is called from within the constructor to initialize
23      * the form.
24      * WARNING: Do NOT modify this code. The content of this method is
25      * always
26      * regenerated by the Form Editor.
27      */
28
29     // <editor-fold defaultstate="collapsed" desc="Generated Code">
30     GEN-BEGIN: initComponents
31     private void initComponents() {
32
33         jButtonSiguiente = new javax.swing.JButton();
34         jButtonAnterior = new javax.swing.JButton();
35         jButtonCancelar = new javax.swing.JButton();
36         jLabel1 = new javax.swing.JLabel();
37         jLabel2 = new javax.swing.JLabel();
38         jLabel3 = new javax.swing.JLabel();
39         jButtonNoTerminales = new javax.swing.JButton();
40         jButtonTerminales = new javax.swing.JButton();
41         jScrollPane3 = new javax.swing.JScrollPane();
42         jListNoTerminales = new javax.swing.JList();
43         jScrollPane4 = new javax.swing.JScrollPane();
44         jListTerminales = new javax.swing.JList();
45         jButtonUltimo = new javax.swing.JButton();
46         jButtonPrimero = new javax.swing.JButton();
47
48         setBackground(new java.awt.Color(233, 244, 244));
49
50         jButtonSiguiente.setForeground(new java.awt.Color(33, 77, 72));

```

```

47      jButtonSiguiente.setIcon(new javax.swing.ImageIcon(getClass().
      getResource("/es/uco/simas/resources/siguiente.png"))); //
      NOI18N
48      jButtonSiguiente.setToolTipText("Siguiente");
49      jButtonSiguiente.addActionListener(new java.awt.event.
      ActionListener() {
50          public void actionPerformed(java.awt.event.ActionEvent evt)
          {
51              jButtonSiguienteActionPerformed(evt);
52          }
53      });
54
55      jButtonAnterior.setForeground(new java.awt.Color(33, 77, 72));
56      jButtonAnterior.setIcon(new javax.swing.ImageIcon(getClass().
      getResource("/es/uco/simas/resources/anterior.png"))); //
      NOI18N
57      jButtonAnterior.setToolTipText("Anterior");
58      jButtonAnterior.addActionListener(new java.awt.event.
      ActionListener() {
59          public void actionPerformed(java.awt.event.ActionEvent evt)
          {
60              jButtonAnteriorActionPerformed(evt);
61          }
62      });
63
64      jButtonCancelar.setForeground(new java.awt.Color(33, 77, 72));
65      jButtonCancelar.setText("Cancelar");
66      jButtonCancelar.addActionListener(new java.awt.event.
      ActionListener() {
67          public void actionPerformed(java.awt.event.ActionEvent evt)
          {
68              jButtonCancelarActionPerformed(evt);
69          }
70      });
71
72      jLabel1.setFont(new java.awt.Font("Ubuntu", 1, 18)); // NOI18N
73      jLabel1.setForeground(new java.awt.Color(33, 77, 72));
74      jLabel1.setText("Vocabulario de la gramática");
75
76      jLabel2.setText("Simbolos No Terminales");
77
78      jLabel3.setText("Simbolos Terminales");
79
80      jButtonNoTerminales.setText("Modificar No Terminales");
81      jButtonNoTerminales.addActionListener(new java.awt.event.
      ActionListener() {
82          public void actionPerformed(java.awt.event.ActionEvent evt)
          {
83              jButtonNoTerminalesActionPerformed(evt);
84          }
85      });
86
87      jButtonTerminales.setText("Modificar Terminales");

```

```

88         jButtonTerminales.addActionListener(new java.awt.event.
           ActionListener() {
89             public void actionPerformed(java.awt.event.ActionEvent evt)
               {
90                 jButtonTerminalesActionPerformed(evt);
91             }
92         });
93
94         jScrollPane3.setViewportViewView(jListNoTerminales);
95
96         jScrollPane4.setViewportViewView(jListTerminales);
97
98         jButtonUltimo.setIcon(new javax.swing.ImageIcon(getClass().
           getResource("/es/uco/simas/resources/ultimo.png"))); //
           NOI18N
99         jButtonUltimo.setToolTipText("Ultimo");
100        jButtonUltimo.addActionListener(new java.awt.event.
           ActionListener() {
101            public void actionPerformed(java.awt.event.ActionEvent evt)
               {
102                jButtonUltimoActionPerformed(evt);
103            }
104        });
105
106        jButtonPrimero.setIcon(new javax.swing.ImageIcon(getClass().
           getResource("/es/uco/simas/resources/primero.png"))); //
           NOI18N
107        jButtonPrimero.setToolTipText("Primero");
108        jButtonPrimero.addActionListener(new java.awt.event.
           ActionListener() {
109            public void actionPerformed(java.awt.event.ActionEvent evt)
               {
110                jButtonPrimeroActionPerformed(evt);
111            }
112        });
113
114        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
           this);
115        this.setLayout(layout);
116        layout.setHorizontalGroup(
117            layout.createParallelGroup(javax.swing.GroupLayout.
           Alignment.LEADING)
118            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
           layout.createSequentialGroup()
119                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
           Short.MAX_VALUE)
120                .addComponent(jLabel1)
121                .addGap(188, 188, 188))
122            .addGroup(layout.createSequentialGroup()
123                .addGap(33, 33, 33)
124                .addGroup(layout.createParallelGroup(javax.swing.
           GroupLayout.Alignment.LEADING)
125                    .addGroup(layout.createSequentialGroup()

```

```

126         .addComponent(jButtonCancelar)
127         .addPreferredGap(javax.swing.LayoutStyle.
            ComponentPlacement.RELATED, javax.swing.
            GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
128         .addComponent(jButtonPrimero)
129         .addPreferredGap(javax.swing.LayoutStyle.
            ComponentPlacement.RELATED)
130         .addComponent(jButtonAnterior)
131         .addPreferredGap(javax.swing.LayoutStyle.
            ComponentPlacement.RELATED)
132         .addComponent(jButtonSiguiente)
133         .addPreferredGap(javax.swing.LayoutStyle.
            ComponentPlacement.RELATED)
134         .addComponent(jButtonUltimo)
135         .addGap(57, 57, 57))
136     .addGroup(layout.createSequentialGroup())
137     .addGroup(layout.createParallelGroup(javax.
        swing.GroupLayout.Alignment.LEADING)
        .addComponent(jLabel2)
138        .addComponent(jScrollPane3, javax.swing.
            GroupLayout.PREFERRED_SIZE, 222, javax.
            swing.GroupLayout.PREFERRED_SIZE)
139        .addGroup(layout.createSequentialGroup())
            .addGap(34, 34, 34)
            .addComponent(jButtonNoTerminales)))
140     .addGroup(layout.createParallelGroup(javax.
        swing.GroupLayout.Alignment.LEADING)
141        .addGroup(javax.swing.GroupLayout.Alignment
            .TRAILING, layout.createSequentialGroup
            ())
142        .addPreferredGap(javax.swing.
            LayoutStyle.ComponentPlacement.
            RELATED, javax.swing.GroupLayout.
            DEFAULT_SIZE, Short.MAX_VALUE)
143        .addComponent(jButtonTerminales)
            .addGap(79, 79, 79))
144     .addGroup(layout.createSequentialGroup())
            .addGap(96, 96, 96)
145     .addGroup(layout.createParallelGroup(
        javax.swing.GroupLayout.Alignment.
        LEADING)
146        .addComponent(jScrollPane4, javax.
            swing.GroupLayout.
            PREFERRED_SIZE, 221, javax.
            swing.GroupLayout.
            PREFERRED_SIZE)
147        .addComponent(jLabel3))
148     .addContainerGap(58, Short.MAX_VALUE)))
149 )))
150
151 );
152 layout.setVerticalGroup(
153     layout.createParallelGroup(javax.swing.GroupLayout.
        Alignment.LEADING)
154

```



```

157         .addGroup(layout.createSequentialGroup())
158         .addContainerGap()
159         .addComponent(jLabel1)
160         .addGap(15, 15, 15)
161         .addGroup(layout.createParallelGroup(javax.swing.
            GroupLayout.Alignment.BASELINE)
162         .addComponent(jLabel2)
163         .addComponent(jLabel3))
164         .addPreferredGap(javax.swing.LayoutStyle.
            ComponentPlacement.RELATED)
165         .addGroup(layout.createParallelGroup(javax.swing.
            GroupLayout.Alignment.LEADING)
166         .addGroup(layout.createSequentialGroup()
167         .addComponent(jScrollPane3, javax.swing.
            GroupLayout.PREFERRED_SIZE, 201, javax.
            swing.GroupLayout.PREFERRED_SIZE)
168         .addGap(7, 7, 7)
169         .addComponent(jButtonNoTerminales)
170         .addGap(0, 0, Short.MAX_VALUE))
171         .addGroup(layout.createSequentialGroup()
172         .addComponent(jScrollPane4, javax.swing.
            GroupLayout.PREFERRED_SIZE, 201, javax.
            swing.GroupLayout.PREFERRED_SIZE)
173         .addPreferredGap(javax.swing.LayoutStyle.
            ComponentPlacement.RELATED)
174         .addComponent(jButtonTerminales)
175         .addPreferredGap(javax.swing.LayoutStyle.
            ComponentPlacement.RELATED, 18, Short.
            MAX_VALUE)
176         .addGroup(layout.createParallelGroup(javax.
            swing.GroupLayout.Alignment.LEADING)
177         .addComponent(jButtonAnterior)
178         .addComponent(jButtonSiguiente)
179         .addComponent(jButtonPrimero)
180         .addComponent(jButtonCancelar)
181         .addComponent(jButtonUltimo))
182         .addContainerGap()))
183     );
184 }// </editor-fold>//GEN-END: initComponents
185
186 private void jButtonSiguienteActionPerformed(java.awt.event.
    ActionEvent evt) { //GEN-FIRST:
    event_jButtonSiguienteActionPerformed
187     this.ventanaPadre.cambiarPaso(3);
188 }//GEN-LAST: event_jButtonSiguienteActionPerformed
189
190 private void jButtonCancelarActionPerformed(java.awt.event.
    ActionEvent evt) { //GEN-FIRST:
    event_jButtonCancelarActionPerformed
191     int conf = JOptionPane.showConfirmDialog(null, "¿Desea salir
    de la edicion de la gramática?", "Salir",JOptionPane.
    YES_NO_OPTION);
192

```

```

193         if (conf==0){
194             this.ventanaPadre.finalizarAsistente();
195         }
196     } //GEN-LAST: event_jButtononCancelarActionPerformed
197
198     private void jButtonAnteriorActionPerformed(java.awt.event.
        ActionEvent evt) { //GEN-FIRST:
        event_jButtononAnteriorActionPerformed
199         this.ventanaPadre.cambiarPaso(1);
200     } //GEN-LAST: event_jButtononAnteriorActionPerformed
201
202     private void jButtonNoTerminalesActionPerformed(java.awt.event.
        ActionEvent evt) { //GEN-FIRST:
        event_jButtononNoTerminalesActionPerformed
203         VentanaSimbolosNoTerminales noTerminal = new
            VentanaSimbolosNoTerminales(this.jListNoTerminales.getModel()
                (), this.jListNoTerminales.getModel(), this, null);
204         noTerminal.setVisible(true);
205         noTerminal.setLocationRelativeTo(null);
206     } //GEN-LAST: event_jButtononNoTerminalesActionPerformed
207
208     private void jButtonTerminalesActionPerformed(java.awt.event.
        ActionEvent evt) { //GEN-FIRST:
        event_jButtononTerminalesActionPerformed
209         VentanaSimbolosTerminales terminal = new
            VentanaSimbolosTerminales(this.jListTerminales.getModel(),
                this.jListTerminales.getModel(), this);
210         terminal.setVisible(true);
211         terminal.setLocationRelativeTo(null);
212     } //GEN-LAST: event_jButtononTerminalesActionPerformed
213
214     private void jButtonUltimoActionPerformed(java.awt.event.
        ActionEvent evt) { //GEN-FIRST:
        event_jButtononUltimoActionPerformed
215         this.ventanaPadre.cambiarPaso(4);
216     } //GEN-LAST: event_jButtononUltimoActionPerformed
217
218     private void jButtonPrimeroActionPerformed(java.awt.event.
        ActionEvent evt) { //GEN-FIRST:
        event_jButtononPrimeroActionPerformed
219         this.ventanaPadre.cambiarPaso(1);
220     } //GEN-LAST: event_jButtononPrimeroActionPerformed
221
222     public DefaultListModel getSimbolosTerminales( ) {
223         if (this.jListTerminales.getModel().getSize() != 0) {
224             return (DefaultListModel) this.jListTerminales.getModel();
225         }
226         return null;
227     }
228
229     public DefaultListModel getSimbolosNoTerminales( ) {
230         if (this.jListNoTerminales.getModel().getSize() != 0) {
231             return (DefaultListModel) this.jListNoTerminales.getModel();

```

```

232     }
233     return null;
234 }
235
236 public void asignarListaSimbolosTerminales(DefaultListModel
    modelo) {
237     if(modelo != null)
238         this.jListTerminales.setModel(modelo);
239     else
240         this.jListTerminales = null;
241 }
242
243 public void asignarListaSimbolosNoTerminales( DefaultListModel
    modelo) {
244     if(modelo != null)
245         this.jListNoTerminales.setModel(modelo);
246     else
247         this.jListNoTerminales = null;
248 }
249
250 public DefaultListModel getProducciones() {
251     return this.ventanaPadre.getProducciones();
252 }
253
254 public void setProducciones(DefaultListModel pr){
255     this.ventanaPadre.setProducciones(pr);
256 }
257
258 // Variables declaration – do not modify//GEN-BEGIN:variables
259 private javax.swing.JButton jButtonAnterior;
260 private javax.swing.JButton jButtonCancelar;
261 private javax.swing.JButton jButtonNoTerminales;
262 private javax.swing.JButton jButtonPrimero;
263 private javax.swing.JButton jButtonSiguiente;
264 private javax.swing.JButton jButtonTerminales;
265 private javax.swing.JButton jButtonUltimo;
266 private javax.swing.JLabel jLabel1;
267 private javax.swing.JLabel jLabel2;
268 private javax.swing.JLabel jLabel3;
269 private javax.swing.JList jListNoTerminales;
270 private javax.swing.JList jListTerminales;
271 private javax.swing.JScrollPane jScrollPane3;
272 private javax.swing.JScrollPane jScrollPane4;
273 // End of variables declaration//GEN-END:variables
274 }

```

#### 2.4.14. PanelCreacionGramaticaPaso3.java

```

1 // SimAS / Editor
2 // Panel Creacion Gramática Paso 3

```

```

3
4 package es.uco.simas.editor;
5
6 import javax.swing.JOptionPane;
7 import javax.swing.DefaultListModel;
8
9 /**
10  * @author vanesa
11  */
12 public class PanelCreacionGramaticaPaso3 extends javax.swing.JPanel {
13
14     private VentanaCreacionGramatica ventanaPadre ;
15
16     public PanelCreacionGramaticaPaso3(VentanaCreacionGramatica
17         ventanaPadre) {
18         initComponents();
19         this.ventanaPadre = ventanaPadre;
20     }
21     /**
22      * This method is called from within the constructor to initialize
23      * the form.
24      * WARNING: Do NOT modify this code. The content of this method is
25      * always
26      * regenerated by the Form Editor.
27      */
28
29     // <editor-fold defaultstate="collapsed" desc="Generated Code">
30     // GEN-BEGIN: initComponents
31     private void initComponents() {
32
33         jButtonSiguiente = new javax.swing.JButton();
34         jButtonAnterior = new javax.swing.JButton();
35         jButtonCancelar = new javax.swing.JButton();
36         jLabel1 = new javax.swing.JLabel();
37         jScrollPane1 = new javax.swing.JScrollPane();
38         jListProduccion = new javax.swing.JList();
39         jButtonModificar = new javax.swing.JButton();
40         jButton1 = new javax.swing.JButton();
41         jButton2 = new javax.swing.JButton();
42
43         setBackground(new java.awt.Color(233, 244, 244));
44
45         jButtonSiguiente.setForeground(new java.awt.Color(33, 77, 72));
46         jButtonSiguiente.setIcon(new javax.swing.ImageIcon(getClass().
47             getResource("/es/uco/simas/resources/siguiente.png"))); //
48             NOI18N
49         jButtonSiguiente.setToolTipText("Siguiente");
50         jButtonSiguiente.addActionListener(new java.awt.event.
51             ActionListener() {
52             public void actionPerformed(java.awt.event.ActionEvent evt)
53             {
54                 jButtonSiguienteActionPerformed(evt);
55             }
56         });

```

```

48     });
49
50     jButtonAnterior.setForeground(new java.awt.Color(33, 77, 72));
51     jButtonAnterior.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/anterior.png"))); //
        NOI18N
52     jButtonAnterior.setToolTipText("Anterior");
53     jButtonAnterior.addActionListener(new java.awt.event.
        ActionListener() {
54         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
55             jButtonAnteriorActionPerformed(evt);
56         }
57     });
58
59     jButtonCancelar.setForeground(new java.awt.Color(33, 77, 72));
60     jButtonCancelar.setText("Cancelar");
61     jButtonCancelar.addActionListener(new java.awt.event.
        ActionListener() {
62         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
63             jButtonCancelarActionPerformed(evt);
64         }
65     });
66
67     jLabel1.setFont(new java.awt.Font("Ubuntu", 1, 18)); // NOI18N
68     jLabel1.setForeground(new java.awt.Color(33, 77, 72));
69     jLabel1.setText("Producciones de la gramática");
70
71     jScrollPane1.setViewportView(jListProduccion);
72
73     jButtonModificar.setText("Modificar Producciones");
74     jButtonModificar.addActionListener(new java.awt.event.
        ActionListener() {
75         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
76             jButtonModificarActionPerformed(evt);
77         }
78     });
79
80     jButton1.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/ultimo.png"))); //
        NOI18N
81     jButton1.setToolTipText("Ultimo");
82     jButton1.addActionListener(new java.awt.event.ActionListener()
        {
83         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
84             jButton1ActionPerformed(evt);
85         }
86     });
87

```

```

88      jButton2.setIcon(new javax.swing.ImageIcon(getClass().
      getResource("/es/uco/simas/resources/primero.png"))); //
      NOI18N
89      jButton2.setToolTipText("Primero");
90
91      javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
      this);
92      this.setLayout(layout);
93      layout.setHorizontalGroup(
94          layout.createParallelGroup(javax.swing.GroupLayout.
          Alignment.LEADING)
95          .addGroup(layout.createSequentialGroup()
96              .addGap(42, 42, 42)
97              .addGroup(layout.createParallelGroup(javax.swing.
          GroupLayout.Alignment.LEADING)
98                  .addGroup(layout.createSequentialGroup()
99                      .addComponent(jScrollPane1)
100                     .addGap(42, 42, 42))
101                  .addGroup(layout.createSequentialGroup()
102                      .addComponent(jButtonCancelar)
103                      .addPreferredGap(javax.swing.LayoutStyle.
          ComponentPlacement.RELATED, javax.swing.
          GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
104                      .addComponent(jButton2)
105                      .addPreferredGap(javax.swing.LayoutStyle.
          ComponentPlacement.RELATED)
106                      .addComponent(jButtonAnterior)
107                      .addPreferredGap(javax.swing.LayoutStyle.
          ComponentPlacement.RELATED)
108                      .addComponent(jButtonSiguiente)
109                      .addPreferredGap(javax.swing.LayoutStyle.
          ComponentPlacement.RELATED)
110                      .addComponent(jButton1)
111                      .addGap(46, 46, 46))))
112          .addGroup(layout.createSequentialGroup()
113              .addGroup(layout.createParallelGroup(javax.swing.
          GroupLayout.Alignment.LEADING)
114                  .addGroup(layout.createSequentialGroup()
115                      .addGap(209, 209, 209)
116                      .addComponent(jButtonModificar))
117                  .addGroup(layout.createSequentialGroup()
118                      .addGap(183, 183, 183)
119                      .addComponent(jLabel1)))
120              .addGap(186, 186, 186))
121      );
122      layout.setVerticalGroup(
123          layout.createParallelGroup(javax.swing.GroupLayout.
          Alignment.LEADING)
124          .addGroup(layout.createSequentialGroup()
125              .addGap(19, 19, 19)
126              .addComponent(jLabel1)
127              .addPreferredGap(javax.swing.LayoutStyle.
          ComponentPlacement.RELATED, javax.swing.
          GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

```

```

128         .addComponent(jScrollPane1 , javax.swing.GroupLayout.
            PREFERRED_SIZE, 188, javax.swing.GroupLayout.
            PREFERRED_SIZE)
129         .addPreferredGap(javax.swing.LayoutStyle.
            ComponentPlacement.UNRELATED)
130         .addComponent(jButtonModificar)
131         .addPreferredGap(javax.swing.LayoutStyle.
            ComponentPlacement.UNRELATED)
132         .addGroup(layout.createParallelGroup(javax.swing.
            GroupLayout.Alignment.LEADING)
133             .addGroup(javax.swing.GroupLayout.Alignment.
                TRAILING, layout.createParallelGroup(javax.
                    swing.GroupLayout.Alignment.LEADING)
134                 .addComponent(jButtonSiguiente , javax.swing.
                    GroupLayout.Alignment.TRAILING)
135                 .addComponent(jButtonAnterior , javax.swing.
                    GroupLayout.Alignment.TRAILING)
136                 .addGroup(layout.createParallelGroup(javax.
                    swing.GroupLayout.Alignment.BASELINE)
137                     .addComponent(jButtonCancelar)
138                     .addComponent(jButton2)))
139             .addComponent(jButton1 , javax.swing.GroupLayout.
                Alignment.TRAILING))
140         .addGap(40, 40, 40))
141     );
142 }// </editor-fold>//GEN-END: initComponents
143
144 private void jButtonSiguienteActionPerformed(java.awt.event.
    ActionEvent evt) { //GEN-FIRST:
    event_jButtonSiguienteActionPerformed
145         this.ventanaPadre.cambiarPaso(4);
146 } //GEN-LAST: event_jButtonSiguienteActionPerformed
147
148 private void jButtonCancelarActionPerformed(java.awt.event.
    ActionEvent evt) { //GEN-FIRST:
    event_jButtonCancelarActionPerformed
149         int conf = JOptionPane.showConfirmDialog(null, "¿Desea salir
            de la edicion de la gramática?", "Salir",JOptionPane.
            YES.NO.OPTION);
150
151         if (conf==0)
152             this.ventanaPadre.finalizarAsistente();
153 } //GEN-LAST: event_jButtonCancelarActionPerformed
154
155 private void jButtonAnteriorActionPerformed(java.awt.event.
    ActionEvent evt) { //GEN-FIRST:
    event_jButtonAnteriorActionPerformed
156         this.ventanaPadre.cambiarPaso(2);
157 } //GEN-LAST: event_jButtonAnteriorActionPerformed
158
159 private void jButtonModificarActionPerformed(java.awt.event.
    ActionEvent evt) { //GEN-FIRST:
    event_jButtonModificarActionPerformed

```

```

160         VentanaProducciones producciones = new VentanaProducciones(this
161             , this.jListProduccion.getModel());
162         producciones.setVisible(true);
163         producciones.setLocationRelativeTo(null);
164     } //GEN-LAST:event_jButtonModificarActionPerformed
165
166     private void jButton1ActionPerformed(java.awt.event.ActionEvent evt
167         ) { //GEN-FIRST:event_jButton1ActionPerformed
168         this.ventanaPadre.cambiarPaso(4);
169     } //GEN-LAST:event_jButton1ActionPerformed
170
171     public DefaultListModel getSimbolosTerminales() {
172         return this.ventanaPadre.getSimbolosTerminales();
173     }
174
175     public DefaultListModel getSimbolosNoTerminales() {
176         return this.ventanaPadre.getSimbolosNoTerminales();
177     }
178
179     public void asignarProducciones(DefaultListModel producciones)
180     {
181         if(producciones != null){
182             this.jListProduccion.setModel(producciones);
183         }else
184             this.jListProduccion = null;
185     }
186
187     public DefaultListModel getProducciones() {
188         DefaultListModel producciones= new DefaultListModel();
189         if(this.jListProduccion.getModel().getSize()!=0) {
190             producciones=(DefaultListModel) this.jListProduccion.
191                 getModel();
192             return producciones;
193         }
194         return null;
195     }
196
197     // Variables declaration - do not modify//GEN-BEGIN:variables
198     private javax.swing.JButton jButton1;
199     private javax.swing.JButton jButton2;
200     private javax.swing.JButton jButtonAnterior;
201     private javax.swing.JButton jButtonCancelar;
202     private javax.swing.JButton jButtonModificar;
203     private javax.swing.JButton jButtonSiguiente;
204     private javax.swing.JLabel jLabel1;
205     private javax.swing.JList jListProduccion;
206     private javax.swing.JScrollPane jScrollPane1;
207     // End of variables declaration//GEN-END:variables

```



## 2.4.15. PanelCreacionGramaticaPaso4.java

```

1 // SimAS / Editor
2 // Panel Creacion Gramática Paso 4
3
4 package es.uco.simas.editor;
5
6 import javax.swing.JOptionPane;
7 import javax.swing.DefaultComboBoxModel;
8 import javax.swing.DefaultListModel;
9
10 /**
11  * @author vanesa
12  */
13 public class PanelCreacionGramaticaPaso4 extends javax.swing.JPanel {
14
15     private VentanaCreacionGramatica ventanaPadre ;
16     DefaultListModel noTerminales = new DefaultListModel();
17     String simboloInicial = null;
18
19     public PanelCreacionGramaticaPaso4(VentanaCreacionGramatica
20         ventanaPadre) {
21         super();
22         initComponents();
23         this.ventanaPadre =ventanaPadre;
24     }
25
26     public String getSimboloInicial( ) {
27         if (this.jComboBox1.getModel().getSize()==0) {
28             return null;
29         }
30         if (this.jComboBox1.getSelectedIndex() == -1 && this.jComboBox1.
31             getModel().getSize()!=0) {
32             return this.jComboBox1.getModel().getElementAt(0).toString
33                 ();
34         }
35         if (this.jComboBox1.getSelectedIndex() != -1 && this.jComboBox1.
36             getModel().getSize()!=0) {
37             return this.jComboBox1.getSelectedItem().toString();
38         }
39         return null;
40     }
41
42     public void setSimboloInicial(String inicial){
43         this.simboloInicial= inicial;
44     }
45
46     public void setSimbolosNoTerminales( DefaultListModel
47         simbolosNoTerminales) {
48         this.noTerminales = simbolosNoTerminales;
49         String [] combo = new String[simbolosNoTerminales.getSize()];

```

```

47     int i= 0;
48     int inicial=-1;
49
50     while(true) {
51         if(i >= simbolosNoTerminales.getSize()) {
52             break;
53         }
54         if(i < simbolosNoTerminales.getSize()) {
55             combo[i] = simbolosNoTerminales.getElementAt(i).
                    toString();
56             if(combo[i].equals(this.simboloInicial))
57                 inicial=i;
58         }
59         i = i+1;
60     }
61     DefaultComboBoxModel nt = new DefaultComboBoxModel(combo);
62     this.jComboBox1.setModel(nt);
63     if(inicial != -1)
64         this.jComboBox1.setSelectedIndex(inicial);
65 }
66
67 /**
68  * This method is called from within the constructor to initialize
        the form.
69  * WARNING: Do NOT modify this code. The content of this method is
        always
70  * regenerated by the Form Editor.
71  */
72
73 // <editor-fold defaultstate="collapsed" desc="Generated Code">//
        GEN-BEGIN: initComponents
74 private void initComponents() {
75
76     buttonGroup1 = new javax.swing.ButtonGroup();
77     jButtonFin = new javax.swing.JButton();
78     jButtonAnterior = new javax.swing.JButton();
79     jButtonCancelar = new javax.swing.JButton();
80     jLabel2 = new javax.swing.JLabel();
81     jComboBox1 = new javax.swing.JComboBox();
82     jButtonPrimero = new javax.swing.JButton();
83
84     setBackground(new java.awt.Color(233, 244, 244));
85     setToolTipText("Selección Símbolo Inicial");
86
87     jButtonFin.setFont(new java.awt.Font("Ubuntu", 1, 15)); //
        NOI18N
88     jButtonFin.setForeground(new java.awt.Color(33, 77, 72));
89     jButtonFin.setText("Finalizar");
90     jButtonFin.addActionListener(new java.awt.event.ActionListener
        () {
91         public void actionPerformed(java.awt.event.ActionEvent evt)
        {
92             jButtonFinActionPerformed(evt);

```

```

93         }
94     });
95
96     jButtonAnterior.setForeground(new java.awt.Color(33, 77, 72));
97     jButtonAnterior.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/anterior.png"))); //
        NOI18N
98     jButtonAnterior.setToolTipText("Anterior");
99     jButtonAnterior.addActionListener(new java.awt.event.
        ActionListener() {
100         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
101             jButtonAnteriorActionPerformed(evt);
102         }
103     });
104
105     jButtonCancelar.setForeground(new java.awt.Color(33, 77, 72));
106     jButtonCancelar.setText("Cancelar");
107     jButtonCancelar.addActionListener(new java.awt.event.
        ActionListener() {
108         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
109             jButtonCancelarActionPerformed(evt);
110         }
111     });
112
113     jLabel2.setFont(new java.awt.Font("Ubuntu", 1, 18)); // NOI18N
114     jLabel2.setForeground(new java.awt.Color(33, 77, 72));
115     jLabel2.setText("Seleccione el símbolo Inicial de la
        Gramática:");
116
117     jComboBox1.addActionListener(new java.awt.event.ActionListener
        () {
118         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
119             jComboBox1ActionPerformed(evt);
120         }
121     });
122
123     jButtonPrimero.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/primero.png"))); //
        NOI18N
124     jButtonPrimero.setToolTipText("Primero");
125     jButtonPrimero.addActionListener(new java.awt.event.
        ActionListener() {
126         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
127             jButtonPrimeroActionPerformed(evt);
128         }
129     });
130
131     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
        this);

```



```

167         .addComponent(jButtonPrimero, javax.swing.
        GroupLayout.DEFAULT_SIZE, javax.swing.
        GroupLayout.DEFAULT_SIZE, Short.
        MAX_VALUE)
168         .addComponent(jButtonFin, javax.swing.
        GroupLayout.DEFAULT_SIZE, javax.swing.
        GroupLayout.DEFAULT_SIZE, Short.
        MAX_VALUE))
169         .addGap(40, 40, 40))
170     .addGroup(javax.swing.GroupLayout.Alignment.
        TRAILING, layout.createSequentialGroup())
171     .addComponent(jButtonCancelar)
172     .addGap(37, 37, 37)))
173     );
174 }// </editor-fold> //GEN-END: initComponents
175
176 private void jButtonFinActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST: event_jButtonFinActionPerformed
177     this.ventanaPadre.finalizarAsistente();
178 } //GEN-LAST: event_jButtonFinActionPerformed
179
180 private void jButtonCancelarActionPerformed(java.awt.event.
    ActionEvent evt) { //GEN-FIRST:
    event_jButtonCancelarActionPerformed
181     int conf = JOptionPane.showConfirmDialog(null, "¿Desea salir
        de la edicion de la gramática?", "Salir", JOptionPane.
        YES.NO_OPTION);
182
183     if (conf == 0)
184         this.ventanaPadre.finalizarAsistente(); //dispose();
185 } //GEN-LAST: event_jButtonCancelarActionPerformed
186
187 private void jButtonAnteriorActionPerformed(java.awt.event.
    ActionEvent evt) { //GEN-FIRST:
    event_jButtonAnteriorActionPerformed
188
189     this.ventanaPadre.cambiarPaso(3);
190 } //GEN-LAST: event_jButtonAnteriorActionPerformed
191
192 private void jComboBox1ActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST: event_jComboBox1ActionPerformed
193     // TODO add your handling code here:
194 } //GEN-LAST: event_jComboBox1ActionPerformed
195
196 private void jButtonPrimeroActionPerformed(java.awt.event.
    ActionEvent evt) { //GEN-FIRST:
    event_jButtonPrimeroActionPerformed
197     this.ventanaPadre.cambiarPaso(1);
198 } //GEN-LAST: event_jButtonPrimeroActionPerformed
199
200 // Variables declaration - do not modify //GEN-BEGIN: variables
201 private javax.swing.ButtonGroup buttonGroup1;
202 private javax.swing.JButton jButtonAnterior;

```

```

203 private javax.swing.JButton jButtonCancelar;
204 private javax.swing.JButton jButtonFin;
205 private javax.swing.JButton jButtonPrimero;
206 private javax.swing.JComboBox jComboBox1;
207 private javax.swing.JLabel jLabel2;
208 // End of variables declaration//GEN-END:variables
209 }

```

### 2.4.16. ParteAccion.java

```

1 //SimAS / Editor
2 // PateAccion
3
4 package es.uco.simas.editor;
5
6 import es.uco.simas.util.gramatica.Gramatica;
7 import es.uco.simas.util.gramatica.Produccion;
8 import es.uco.simas.util.gramatica.Terminal;
9 import java.util.ArrayList;
10 import javax.swing.table.DefaultTableModel;
11
12 /**
13  * @author vanesa
14  */
15 public class ParteAccion {
16     ArrayList<FuncionError> funError = new ArrayList();
17     private DefaultTableModel matrizAccion ;
18     Gramatica gramatica;
19
20     public ParteAccion(Gramatica gramatica){
21         this.gramatica = gramatica;
22     }
23
24     void construir(int met){
25         int i =0;
26         ColCanLR0 coleccion = new ColCanLR0();
27         ColCanLR1 col1 = new ColCanLR1();
28         ColCanLALR col2 = new ColCanLALR();
29         if(met == 1)
30             coleccion = this.gramatica.getColeccionLR0();
31         if(met == 2)
32             col1 = this.gramatica.getColeccionLR1();
33         if(met == 3)
34             col2 = this.gramatica.getColeccionLALR();
35
36         this.matrizAccion =new DefaultTableModel();
37         if(met == 3){
38             this.matrizAccion.addColumn("Estados Antiguos");
39             this.matrizAccion.addColumn("Estados Nuevos");
40         }else{

```

```

41         this.matrizAccion.addColumn("Estados");
42     }
43     //Se anaden los simbolos terminales en las columnas
44     while(i < this.gramatica.getTerm().size()){
45         this.matrizAccion.addColumn(this.gramatica.getTerm().get(i)
46             .getNombre());
47         i++;
48     }
49     this.matrizAccion.addColumn("$");
50
51     if(met == 1){ //SLR
52         //Se anaden los estados en las filas
53         i=0;
54         while(i < this.gramatica.getColeccionLR0().
55             getConjElementosLR0().size()){
56             Object [] linea = new Object[] {
57                 i
58             };
59             this.matrizAccion.addRow(linea);
60             i++;
61         }
62
63         //Los anaden los desplazamientos
64         i=0;
65         while(i < coleccion.getConjElementosLR0().size()){
66             int fila = coleccion.getConjElementosLR0().get(i).col;
67             int columna = 0;
68             int j=0;
69             int encontrado =0;
70             while(j < this.matrizAccion.getColumnCount()){
71                 if(coleccion.getConjElementosLR0().get(i).simbolo.
72                     equals(this.matrizAccion.getColumnName(j))){
73                     columna = j;
74                     encontrado = 1;
75                     break;
76                 }else{
77                     j++;
78                 }
79             }
80             if(encontrado == 1){
81                 if(this.matrizAccion.getValueAt(fila , columna)==
82                     null){
83                     this.matrizAccion.setValueAt("d"+coleccion.
84                         getConjElementosLR0().get(i).getI(), fila ,
85                         columna);
86                 }else{
87                     String valor = this.matrizAccion.getValueAt(
88                         fila , columna)+" , d"+coleccion.
89                         getConjElementosLR0().get(i).getI();
90                     this.matrizAccion.setValueAt("conf. "+valor ,
91                         fila , columna);
92                 }
93             }
94             i++;
95         }
96     }

```

```

85     }
86     i=0;
87     while(i < coleccion.getConj2().size()){
88         int fila = coleccion.getConj2().get(i).col;
89         int columna = 0;
90         int j=0;
91         int encontrado =0;
92         while(j < this.matrizAccion.getColumnCount()){
93             if(coleccion.getConj2().get(i).simbolo.equals(this.
94                 matrizAccion.getColumnName(j))){
95                 columna = j;
96                 encontrado = 1;
97                 break;
98             }else
99                 j++;
100         }
101         if(encontrado == 1){
102             if(this.matrizAccion.getValueAt(fila , columna) ==
103                 null){
104                 this.matrizAccion.setValueAt("d"+coleccion.
105                     getConj2().get(i).getI() , fila , columna);
106             }else{
107                 String valor = this.matrizAccion.getValueAt(
108                     fila , columna)+" , d"+coleccion.getConj2().
109                     get(i).getI();
110                 this.matrizAccion.setValueAt("conf. "+valor ,
111                     fila , columna);
112             }
113         }
114         i++;
115     }
116     this.reducciones(met);
117 }
118
119 if(met == 2){    //LR-canónico
120     //Se anaden los estados en las filas
121     i=0;
122     while(i < this.gramatica.getColeccionLR1().
123         getConjElementosLR1().size()){
124         Object [] linea = new Object[] {
125             i
126         };
127         this.matrizAccion.addRow(linea);
128         i++;
129     }
130
131     //Los anaden los desplazamientos
132     i=0;
133     while(i < coll.getConjElementosLR1().size()){
134         int fila = coll.getConjElementosLR1().get(i).col;
135         int columna = 0;
136         int j=0;

```



```

131         int encontrado =0;
132         while(j < this.matrizAccion.getColumnCount()){
133             if(col1.getConjElementosLR1().get(i).simbolo.equals
134                 (this.matrizAccion.getColumnName(j))){
135                 columna = j;
136                 encontrado = 1;
137                 break;
138             }else
139                 j++;
140         }
141         if(encontrado == 1){
142             if(this.matrizAccion.getValueAt(fila , columna)==
143                 null){
144                 this.matrizAccion.setValueAt("d"+col1.
145                     getConjElementosLR1().get(i).getI() , fila ,
146                     columna);
147             }else{
148                 String valor = this.matrizAccion.getValueAt(
149                     fila , columna)+" , d"+col1.
150                     getConjElementosLR1().get(i).getI();
151                 this.matrizAccion.setValueAt("conf. "+valor ,
152                     fila , columna);
153             }
154         }
155         i++;
156     }
157     i=0;
158     while(i < col1.getConj2().size()){
159         int fila = col1.getConj2().get(i).col;
160         int columna = 0;
161         int j=0;
162         int encontrado =0;
163         while(j < this.matrizAccion.getColumnCount()){
164             if(col1.getConj2().get(i).simbolo.equals(this.
165                 matrizAccion.getColumnName(j))){
166                 columna = j;
167                 encontrado = 1;
168                 break;
169             }else
170                 j++;
171         }
172         if(encontrado == 1){
173             if(this.matrizAccion.getValueAt(fila , columna) ==
174                 null)
175                 this.matrizAccion.setValueAt("d"+col1.getConj2
176                     ().get(i).getI() , fila , columna);
177             else{
178                 String valor = this.matrizAccion.getValueAt(
179                     fila , columna)+" , d"+col1.getConj2().get(i)
180                     .getI();
181                 this.matrizAccion.setValueAt("conf. "+valor ,
182                     fila , columna);
183             }
184         }
185     }

```

```

171         }
172         i++;
173     }
174     this.reducciones(met);
175 }
176 if(met == 3){ //LALR
177     //Se anaden los estados en las filas
178     i=0;
179     while(i < this.gramatica.getColeccionLALR().
180         getConjElementosLALR().size()){
181         if(this.gramatica.getColeccionLALR().
182             getConjElementosLALR().get(i).getV() != -1){
183             Object [] linea = new Object [] {
184                 this.gramatica.getColeccionLALR().
185                     getConjElementosLALR().get(i).getI()+"-"+
186                     this.gramatica.getColeccionLALR().
187                         getConjElementosLALR().get(i).getV(),
188                 i
189             };
190             this.matrizAccion.addRow(linea);
191         }else{
192             Object [] linea2 = new Object [] {
193                 this.gramatica.getColeccionLALR().
194                     getConjElementosLALR().get(i).getI(),
195                 i
196             };
197             this.matrizAccion.addRow(linea2);
198         }
199         i++;
200     }
201     //Se anaden los desplazamientos
202     i=0;
203     while(i < col2.getConjElementosLALR().size()){
204         int fila = col2.getConjElementosLALR().get(i).col;
205         int columna = 0;
206         int j=0;
207         int encontrado =0;
208         while(j < this.matrizAccion.getColumnCount()){
209             if(col2.getConjElementosLALR().get(i).simbolo.
210                 equals(this.matrizAccion.getColumnName(j))){
211                 columna = j;
212                 encontrado = 1;
213                 break;
214             }else
215                 j++;
216         }
217         if(encontrado == 1){
218             if(this.matrizAccion.getValueAt(fila, columna) ==
219                 null){
220                 if(col2.getConjElementosLALR().get(i).getV() !=
221                     -1){

```

```

214         this.matrizAccion.setValueAt("d"+col2.
            getConjElementosLALR().get(i).getI()+"-
            "+col2.getConjElementosLALR().get(i).
            getV(), fila, columna);
215     }else{
216         this.matrizAccion.setValueAt("d"+col2.
            getConjElementosLALR().get(i).getI(),
            fila, columna);
217     }
218 }else{
219     String valor= "";
220     if( col2.getConjElementosLALR().get(i).getV() !=
        -1){
221         valor = this.matrizAccion.getValueAt(fila,
            columna)+" , d"+col2.
            getConjElementosLALR().get(i).getI()+"-
            "+col2.getConjElementosLALR().get(i).
            getV();
222     }else{
223         valor = this.matrizAccion.getValueAt(fila,
            columna)+" , d"+col2.
            getConjElementosLALR().get(i).getI();
224     }
225     this.matrizAccion.setValueAt("conf. "+valor,
        fila, columna);
226 }
227 }
228 i++;
229 }
230 i=0;
231 while(i < col2.getConj2().size()){
232     int w=0;
233     int fila = -1;
234     while(w < col2.getConjElementosLALR().size()){
235         if( col2.getConjElementosLALR().get(w).getI() ==
            col2.getConj2().get(i).col){
236             fila = w;
237             break;
238         }
239         w++;
240     }
241     if( fila == -1){
242         w=0;
243         while(w < col2.getConjElementosLALR().size()){
244             if( col2.getConjElementosLALR().get(w).getV() ==
                col2.getConj2().get(i).col){
245                 fila = w;
246                 break;
247             }
248             w++;
249         }
250     }
251     //int fila = col2.getConj2().get(i).col;

```

```

252         int columna = 0;
253         int j=0;
254         int encontrado =0;
255         while(j < this.matrizAccion.getColumnCount()){
256             if(col2.getConj2().get(i).simbolo.equals(this.
                matrizAccion.getColumnName(j))){
257                 columna = j;
258                 encontrado = 1;
259                 break;
260             }else
261                 j++;
262         }
263         if(encontrado == 1){
264             System.out.println(fila + " " + columna);
265             if(this.matrizAccion.getValueAt(fila , columna) ==
                null){
266                 if(col2.getConj2().get(i).getV() != -1){
267                     this.matrizAccion.setValueAt("d"+col2.
                        getConj2().get(i).getI()+"-"+col2.
                        getConj2().get(i).getV(), fila , columna
                    );
268                 }else{
269                     this.matrizAccion.setValueAt("d"+col2.
                        getConj2().get(i).getI(), fila , columna
                    );
270                 }
271             }else{
272                 String valor= "";
273                 if(col2.getConj2().get(i).getV() != -1){
274                     valor = this.matrizAccion.getValueAt(fila ,
                        columna)+" , d"+col2.getConj2().get(i).
                        getI()+"-"+col2.getConj2().get(i).getV
                        ();
275                 }else{
276                     valor = this.matrizAccion.getValueAt(fila ,
                        columna)+" , d"+col2.getConj2().get(i).
                        getI();
277                 }
278                 this.matrizAccion.setValueAt("conf. "+valor ,
                    fila , columna);
279             }
280         }
281         i++;
282     }
283     this.reducciones(met);
284 }
285 }
286 }
287
288 public void reducciones(int met){
289     ArrayList<ConjElementosLR0> coleccion = new ArrayList();
290     ArrayList<ConjElementosLR1> col1 = new ArrayList();
291     ArrayList<ConjElementosLR1> col2 = new ArrayList();

```

```

292     if (met == 1)
293         coleccion = this.gramatica.getColeccionLR0().
                getConjElementosLR0();
294     if (met == 2)
295         col1 = this.gramatica.getColeccionLR1().getConjElementosLR1
                ();
296     if (met == 3)
297         col2 = this.gramatica.getColeccionLALR().
                getConjElementosLALR();
298
299     ArrayList<Produccion> produc = new ArrayList();
300     ArrayList<Integer> estado = new ArrayList();
301     ArrayList<Terminal> siguientes = new ArrayList<>();
302     int i=0;
303     int v=0;
304     if (met == 1){
305         while(i < coleccion.size()){
306             int j=0;
307             while(j < coleccion.get(i).getElementosLR0().size()){
308                 if (coleccion.get(i).getElementosLR0().get(j).
                        getPosicion() == coleccion.get(i).
                        getElementosLR0().get(j).getProduc().getConsec
                        ().size()){
309                     estado.add(coleccion.get(i).getI());
310                     produc.add(coleccion.get(i).getElementosLR0().
                                get(j).getProduc());
311                     v++;
312                 }
313                 j++;
314             }
315             i++;
316         }
317
318         i=0;
319         while(i < estado.size()){
320             if (estado.get(i) == 1 && produc.get(i).getAntec().
                    getSimboloNT().getNombre().equals(this.gramatica.
                    getSimbInicial()+" ")){
321                 this.matrizAccion.setValueAt("Acepta", 1, this.
                    matrizAccion.getColumnCount()-1);
322             } else {
323                 int m=0;
324                 //Conjunto siguiente del antecedente
325                 while(m < this.gramatica.getNoTerm().size()){
326                     siguientes = new ArrayList();
327                     if (this.gramatica.getNoTerm().get(m).getNombre
                        ().equals(produc.get(i).getAntec().
                        getSimboloNT().getNombre())){
328                         siguientes = this.gramatica.getNoTerm().get
                            (m).getSiguietes();
329                         break;
330                     } else
331                         m++;

```

```

332     }
333     int j=1;
334     while(j < this.matrizAccion.getColumnCount()){
335         int k=0;
336         while(k < siguientes.size()){
337             if(this.matrizAccion.getColumnName(j).
                 equals(siguientes.get(k).getNombre())){
338                 int n=1;
339                 int pr =0;
340                 //buscar el numero de la produccion
341                 while(n < this.gramatica.getPr().size()
                    ){
342                     pr=0;
343                     int iguales =1;
344                     if(this.gramatica.getPr().get(n).
                        getAntec().getSimboloNT().
                        getNombre().equals(produc.get(i)
                        ).getAntec().getSimboloNT().
                        getNombre())){
345                         int p=0;
346                         if(this.gramatica.getPr().get(n)
                            .getConsec().size() !=
                            produc.get(i).getConsec().
                            size()){
347                             iguales =0;
348                         }else{
349                             while(p < this.gramatica.
                                getPr().get(n).
                                getConsec().size()){
350                                 if(!this.gramatica.
                                    getPr().get(n).
                                    getConsec().get(p).
                                    getNombre().equals(
                                    produc.get(i).
                                    getConsec().get(p).
                                    getNombre())){
351                                     iguales =0;
352                                     break;
353                                 }else
354                                     p++;
355                             }
356                         }
357                     if(iguales == 1){
358                         pr = n;
359                         break;
360                     }
361                 }
362                 n++;
363             }
364         if(pr !=0){
365             if(this.matrizAccion.getValueAt(
                estado.get(i), j) == null)

```

```

366         this.matrizAccion.setValueAt("r
367         "+pr, estado.get(i), j);
368     else {
369         String valor = this.
            matrizAccion.getValueAt(
                estado.get(i),j)+" , r"+pr;
            this.matrizAccion.setValueAt("
            conf. "+valor, estado.get(i
            ), j);
370     }
371     break;
372 }
373 }
374 k++;
375 }
376 j++;
377 }
378 }
379 i++;
380 }
381 }
382
383 if (met == 2){
384     ArrayList<String> anticipacion = new ArrayList();
385     while(i < coll.size()){
386         int j=0;
387         while(j < coll.get(i).getElementosLR1().size()){
388             //Elementos con el punto al final
389             if (coll.get(i).getElementosLR1().get(j).getPosicion
                () == coll.get(i).getElementosLR1().get(j).
                getProduc().getConsec().size()){
390                 int est = coll.get(i).getI();
391                 Produccion prod = coll.get(i).getElementosLR1()
                    .get(j).getProduc();
392                 anticipacion = coll.get(i).getElementosLR1().
                    get(j).getAnticipacion();
393                 int n=1;
394                 int pr =0;
395                 //buscar el numero de la produccion
396                 while(n < this.gramatica.getPr().size()){
397                     pr=0;
398                     int iguales =1;
399                     if (this.gramatica.getPr().get(n).getAntec()
                        .getSimboloNT().getNombre().equals(prod
                        .getAntec().getSimboloNT().getNombre())
                        ){
400                         int p=0;
401                         while(p < this.gramatica.getPr().get(n)
                            .getConsec().size()){
402                             if (!this.gramatica.getPr().get(n).
                                getConsec().get(p).getNombre().
                                equals(prod.getConsec().get(p).
                                getNombre())){

```

```

403         iguales =0;
404         break;
405     }else
406         p++;
407     }
408     if(iguales == 1){
409         pr = n;
410         break;
411     }
412 }
413 n++;
414 }
415 if(est == 1 && prod.getAntec().getSimboloNT().
    getNombre().equals(this.gramatica.
    getSimbInicial()+" ")){
416     this.matrizAccion.setValueAt("Aceptar", i,
        this.matrizAccion.getColumnCount()-1);
417 }else{
418
419     v=1;
420
421     while(v < this.matrizAccion.getColumnCount
        ()){
422         int w = 0;
423         while(w < anticipacion.size()){
424             if(anticipacion.get(w).equals(this.
                matrizAccion.getColumnName(v))){
425
426                 if(pr !=0){
427                     if(this.matrizAccion.
                        getValueAt(est, v) ==
                        null)
428                         this.matrizAccion.
                            setValueAt("r"+pr,
                                est, v);
429                     else{
430                         String valor = this.
                            matrizAccion.
                                getValueAt(est,v)+"
                                , r"+pr;
431                         this.matrizAccion.
                            setValueAt("conf. "
                                +valor, est, v);
432                     }
433                 }
434             }
435             w++;
436         }
437         v++;
438     }
439 }
440 }

```



```

441         j++;
442     }
443     i++;
444 }
445 }
446
447 if (met == 3) {
448     ArrayList<String> anticipacion = new ArrayList();
449     while (i < col2.size()) {
450         int j=0;
451         while (j < col2.get(i).getElementosLR1().size()) {
452             //Elementos con el punto al final
453             if (col2.get(i).getElementosLR1().get(j).getPosicion()
                == col2.get(i).getElementosLR1().get(j).
                getProduc().getConsec().size()) {
454                 int est = col2.get(i).getY();
455                 Produccion prod = col2.get(i).getElementosLR1()
                    .get(j).getProduc();
456                 anticipacion = col2.get(i).getElementosLR1().
                    get(j).getAnticipacion();
457                 int n=1;
458                 int pr =0;
459                 //buscar el numero de la produccion
460                 while (n < this.gramatica.getPr().size()) {
461                     pr=0;
462                     int iguales =1;
463                     if (this.gramatica.getPr().get(n).getAntec()
                        .getSimboloNT().getNombre().equals(prod
                        .getAntec().getSimboloNT().getNombre())
                    ) {
464                         int p=0;
465                         while (p < this.gramatica.getPr().get(n)
                            .getConsec().size()) {
466                             if (!this.gramatica.getPr().get(n).
                                getConsec().get(p).getNombre().
                                equals(prod.getConsec().get(p).
                                getNombre())) {
467                                 iguales =0;
468                                 break;
469                             } else
470                                 p++;
471                         }
472
473                         if (iguales == 1) {
474                             pr = n;
475                             break;
476                         }
477                     }
478                     n++;
479                 }
480                 if (est == 1 && prod.getAntec().getSimboloNT().
                    getNombre().equals(this.gramatica.
                    getSimbInicial()+"'")) {

```

```

481         this.matrizAccion.setValueAt("Aceptar", i ,
482         this.matrizAccion.getColumnCount()-1);
483     }else{
484         v=1;
485         while(v < this.matrizAccion.getColumnCount
486         ()){
487             int w = 0;
488             while(w < anticipacion.size()){
489                 if(anticipacion.get(w).equals(this.
490                 matrizAccion.getColumnName(v))
491                 {
492                     if(pr !=0){
493                         if(this.matrizAccion.
494                         getValueAt(est , v) ==
495                         null)
496                             this.matrizAccion.
497                             setValueAt("r"+pr ,
498                             est , v);
499                     }
500                     else{
501                         String valor = this.
502                         matrizAccion.
503                         getValueAt(est ,v)+"
504                         , r"+pr;
505                         this.matrizAccion.
506                         setValueAt("conf. "
507                         +valor , est , v);
508                     }
509                 }
510             }
511             w++;
512         }
513         v++;
514     }
515     j++;
516 }
517 i++;
518 }
519 }
520
521 public DefaultTableModel getMatrizAccion() {
522     return matrizAccion;
523 }
524
525 public void crearFunError(FuncionError fun){
526     this.funError.add(fun);
527 }

```

```

521     public ArrayList<FuncionError> getFunError() {
522         return funError;
523     }
524
525     public void setFunError(ArrayList<FuncionError> funcion){
526         this.funError = funcion;
527     }
528 }

```

### 2.4.17. ParteIrA.java

```

1  //SimAS / Editor
2  // Parte ir_a
3
4  package es.uco.simas.editor;
5
6  import es.uco.simas.util.gramatica.Gramatica;
7  import javax.swing.table.DefaultTableModel;
8
9  /**
10   * @author vanesa
11   */
12  public class ParteIrA {
13      private DefaultTableModel matrizIrA ;
14      Gramatica gramatica;
15
16      public ParteIrA(Gramatica gramatica){
17          this.gramatica = gramatica;
18      }
19
20      void construir(int met){
21          ColCanLR0 coleccion = new ColCanLR0();
22          ColCanLR1 col1 = new ColCanLR1();
23          ColCanLALR col2 = new ColCanLALR();
24          if(met == 1){
25              coleccion = this.gramatica.getColeccionLR0();
26          }
27          if(met == 2){
28              col1 = this.gramatica.getColeccionLR1();
29          }
30          if(met == 3){
31              col2 = this.gramatica.getColeccionLALR();
32          }
33
34          int i =0;
35          this.matrizIrA =new DefaultTableModel();
36          //Se anaden los simbolos no terminales en las columnas
37          while(i < this.gramatica.getNoTerm().size()){
38              this.matrizIrA.addColumn(this.gramatica.getNoTerm().get(i).
getNombre());

```

```

39         i++;
40     }
41     if (met == 1) {
42         //Se anaden los estados en las filas
43         i=0;
44         while (i < this.gramatica.getColeccionLR0().
45             getConjElementosLR0().size()) {
46             Object [] linea = new Object[] {
47
48                 };
49             this.matrizIrA.addRow(linea);
50             i++;
51         }
52         i=0;
53         while (i < coleccion.getConjElementosLR0().size()) {
54             int fila = coleccion.getConjElementosLR0().get(i).col;
55             int columna = 0;
56             int j=0;
57             int encontrado =0;
58             while (j < this.matrizIrA.getColumnCount()) {
59                 if (coleccion.getConjElementosLR0().get(i).simbolo.
60                     equals(this.matrizIrA.getColumnName(j))) {
61                     columna = j;
62                     encontrado = 1;
63                     break;
64                 } else
65                     j++;
66             }
67             if (encontrado == 1)
68                 this.matrizIrA.setValueAt(coleccion.
69                     getConjElementosLR0().get(i).getI(), fila,
70                     columna);
71
72             i++;
73         }
74         i=0;
75         while (i < coleccion.getConj2().size()) {
76             int fila = coleccion.getConj2().get(i).col;
77             int columna = 0;
78             int j=0;
79             int encontrado =0;
80             while (j < this.matrizIrA.getColumnCount()) {
81                 if (coleccion.getConj2().get(i).simbolo.equals(this.
82                     matrizIrA.getColumnName(j))) {
83                     columna = j;
84                     encontrado = 1;
85                     break;
86                 } else
87                     j++;
88             }
89             if (encontrado == 1) {

```

```

86         this.matrizIrA.setValueAt(coleccion.getConj2().get(
87             i).getI(), fila, columna);
88     }
89     i++;
90 }
91 if(met == 2){
92     //Se anaden los estados en las filas
93     i=0;
94     while(i < this.gramatica.getColeccionLR1().
95         getConjElementosLR1().size()){
96         Object [] linea = new Object[] {
97             };
98         this.matrizIrA.addRow(linea);
99         i++;
100     }
101     i=0;
102     while(i < coll.getConjElementosLR1().size()){
103         int fila = coll.getConjElementosLR1().get(i).col;
104         int columna = 0;
105         int j=0;
106         int encontrado =0;
107         while(j < this.matrizIrA.getColumnCount()){
108             if(coll.getConjElementosLR1().get(i).simbolo.equals
109                 (this.matrizIrA.getUserName(j))){
110                 columna = j;
111                 encontrado = 1;
112                 break;
113             }else
114                 j++;
115         }
116         if(encontrado == 1)
117             this.matrizIrA.setValueAt(coll.getConjElementosLR1
118                 ().get(i).getI(), fila, columna);
119
120         i++;
121     }
122     i=0;
123     while(i < coll.getConj2().size()){
124         int fila = coll.getConj2().get(i).col;
125         int columna = 0;
126         int j=0;
127         int encontrado =0;
128         while(j < this.matrizIrA.getColumnCount()){
129             if(coll.getConj2().get(i).simbolo.equals(this.
130                 matrizIrA.getUserName(j))){
131                 columna = j;
132                 encontrado = 1;
133                 break;
134             }else
135                 j++;
136         }
137     }

```

```

134         if(encontrado == 1){
135             this.matrizIrA.setValueAt(col1.getConj2().get(i).
                getI(), fila, columna);
136         }
137         i++;
138     }
139 }
140
141 if(met == 3){
142     //Se anaden los estados en las filas
143     i=0;
144     while(i < this.gramatica.getColeccionLALR().
        getConjElementosLALR().size()){
145         Object [] linea = new Object[] {
146             };
147         this.matrizIrA.addRow(linea);
148         i++;
149     }
150     i=0;
151     while(i < col2.getConjElementosLALR().size()){
152         int fila = col2.getConjElementosLALR().get(i).col;
153         int columna = 0;
154         int j=0;
155         int encontrado =0;
156         while(j < this.matrizIrA.getColumnCount()){
157             if(col2.getConjElementosLALR().get(i).simbolo.
                equals(this.matrizIrA.getColumnName(j))){
158                 columna = j;
159                 encontrado = 1;
160                 break;
161             }else
162                 j++;
163         }
164         if(encontrado == 1)
165             if(col2.getConjElementosLALR().get(i).getV() != -1)
166                 {
167                     this.matrizIrA.setValueAt(col2.
                        getConjElementosLALR().get(i).getI()+"-"+
                        col2.getConjElementosLALR().get(i).getV(),
                        fila, columna);
167                 }else{
168                     this.matrizIrA.setValueAt(col2.
                        getConjElementosLALR().get(i).getI(), fila,
                        columna);
169                 }
170         i++;
171     }
172     i=0;
173     while(i < col2.getConj2().size()){
174         int w=0;
175         int fila = -1;
176         while(w < col2.getConjElementosLALR().size()){

```

```

177         if ( col2.getConjElementosLALR().get(w).getI() ==
178             col2.getConj2().get(i).col){
179             fila = w;
180             break;
181         }
182         w++;
183     }
184     if ( fila == -1){
185         w=0;
186         while(w < col2.getConjElementosLALR().size()){
187             if ( col2.getConjElementosLALR().get(w).getV() ==
188                 col2.getConj2().get(i).col){
189                 fila = w;
190                 break;
191             }
192             w++;
193         }
194     }
195     int columna = 0;
196     int j=0;
197     int encontrado =0;
198     while(j < this.matrizIrA.getColumnCount()){
199         if ( col2.getConj2().get(i).simbolo.equals(this.
200             matrizIrA.getColumnName(j))){
201             columna = j;
202             encontrado = 1;
203             break;
204         }else
205             j++;
206     }
207     if(encontrado == 1){
208         if ( col2.getConj2().get(i).getV() != -1){
209             this.matrizIrA.setValueAt(col2.getConj2().get(i)
210                 .getI()+"-"+col2.getConj2().get(i).getV(),
211                 fila , columna);
212         }else{
213             this.matrizIrA.setValueAt(col2.getConj2().get(i)
214                 .getI(), fila , columna);
215         }
216     }
217     i++;
218 }
219 }
220 }

public DefaultTableModel getMatrizIrA () {
    return matrizIrA;
}

```

### 2.4.18. TablaLR.java

```
1 // SimAS / Editor
2 // Tabla LR
3
4 package es.uco.simas.editor;
5
6 import es.uco.simas.util.gramatica.Gramatica;
7 import javax.swing.table.DefaultTableModel;
8
9 /**
10  * @author vanesa
11  */
12 public class TablaLR {
13     int metodoAscendente;
14     ParteAccion TAccion;
15     ParteIrA TIrA;
16     DefaultTableModel matrizLR ;
17     DefaultTableModel matrizAccion;
18     DefaultTableModel matrizIra;
19     Gramatica gramatica;
20
21     public TablaLR(Gramatica gramatica){
22         this.gramatica = gramatica;
23     }
24
25     public void construir(int i){
26         this.TAccion = new ParteAccion(this.gramatica);
27         this.TAccion.construir(i);
28         this.TIrA = new ParteIrA(this.gramatica);
29         this.TIrA.construir(i);
30         this.matrizAccion = this.TAccion.getMatrizAccion();
31         this.matrizIra = this.TIrA.getMatrizIrA();
32
33         this.gramatica.setTlr(this);
34     }
35
36     public int getMetodoAscendente() {
37         return metodoAscendente;
38     }
39
40     public void setMetodoAscendente(int metodoAscendente) {
41         this.metodoAscendente = metodoAscendente;
42     }
43
44     public ParteAccion getTAccion() {
45         return TAccion;
46     }
47
48     public void setTAccion(ParteAccion TAccion) {
49         this.TAccion = TAccion;
50     }
51 }
```



```
52     public ParteIrA getTlrA() {
53         return TlrA;
54     }
55
56     public void setTlrA(ParteIrA TlrA) {
57         this.TlrA = TlrA;
58     }
59
60     public DefaultTableModel getMatrizLR() {
61         return matrizLR;
62     }
63 }
```

### 2.4.19. TablaPredictiva.java

```
1  //SimAS / Editor
2  //Tabla Predictiva
3
4  package es.uco.simas.editor;
5
6  import es.uco.simas.util.gramatica.Gramatica;
7  import es.uco.simas.util.gramatica.Terminal;
8  import java.util.ArrayList;
9  import javax.swing.DefaultListModel;
10 import javax.swing.table.DefaultTableModel;
11
12 /**
13  * @author vanesa
14  */
15 public class TablaPredictiva {
16
17     ArrayList<FuncionError> funError = new ArrayList();
18     private DefaultTableModel matrizPred ;
19     Gramatica gramatica;
20
21     public TablaPredictiva(){
22         DefaultTableModel tabla = new DefaultTableModel();
23         this.matrizPred = tabla;
24     }
25
26     public void construir(Gramatica gramatica){
27         int i = 0;
28         int j = 0;
29         int fila = 0;
30         int columna = 0;
31         this.gramatica = gramatica;
32         DefaultListModel produc = gramatica.getProducciones();
33         ArrayList<Terminal> primeros = null;
34         ArrayList<Terminal> siguientes = null;
35     }
```

```

36     this.matrizPred = new DefaultTableModel();
37     this.matrizPred.addColumn("");
38     //Se anaden los simbolos terminales en las columnas
39     while(i < this.gramatica.getTerm().size()){
40         this.matrizPred.addColumn(this.gramatica.getTerm().get(i).
41             getNombre());
42         i++;
43     }
44     this.matrizPred.addColumn("$");
45     //Se anaden los simbolos no terminales en las filas
46     i=0;
47     while(i < this.gramatica.getNoTerm().size()){
48         Object [] linea = new Object[] {
49             this.gramatica.getNoTerm().get(i).getNombre()
50         };
51         this.matrizPred.addRow(linea);
52
53         i++;
54     }
55
56     i=0;
57     while(i < produc.size()){
58         String valor = produc.elementAt(i).toString();
59         String [] separado = valor.split(" ");
60         String antec = separado[0];
61
62         int k=0;
63         while(k < this.gramatica.getNoTerm().size()){
64             if(this.gramatica.getNoTerm().get(k).getNombre().equals
65                 (separado[0])){
66                 fila = k;
67                 break;
68             }
69             k++;
70         }
71
72         if(this.gramatica.isNoTerminal(separado[2])){
73             j=0;
74             while(j < this.gramatica.getNoTerm().size()){
75                 if(this.gramatica.getNoTerm().get(j).getNombre().
76                     equals(separado[2])){
77                     primeros = this.gramatica.getNoTerm().get(j).
78                         getPrimeros();
79                     break;
80                 }else
81                     j++;
82             }
83             if(primeros != null){
84                 j=0;
85                 while(j < primeros.size()){
86                     String nombre = ""+(i+1);
87                     Object objeto = nombre;

```

```

85
86         k=0;
87         while(k < this.gramatica.getTerm().size()){
88             if(this.gramatica.getTerm().get(k).
                getNombre().equals(primeros.get(j).
                getNombre())){
89                 columna = k+1;
90                 break;
91             }
92             k++;
93         }
94         if(columna!=0)
95             setCeldaPredictiva(objeto, fila, columna);
96
97         j++;
98     }
99 }
100 }else{
101     if(this.gramatica.isTerminal(separado[2])){
102         String nombre = ""+(i+1);
103         Object objeto = nombre;
104         k=0;
105         while(k < this.gramatica.getTerm().size()){
106             if(this.gramatica.getTerm().get(k).getNombre().
                equals(separado[2])){
107                 columna = k+1;
108                 break;
109             }
110             k++;
111         }
112         setCeldaPredictiva(objeto, fila, columna);
113
114     }else{ // Si contiene epsilon se utiliza el conj.
115             siguiente
116             if(separado[2].equals("\u03b5")){
117                 j=0;
118                 while(j < this.gramatica.getNoTerm().size()){
119                     if(this.gramatica.getNoTerm().get(j).
                        getNombre().equals(antec)){
120                         siguientes = this.gramatica.getNoTerm().
                            get(j).getSiguientes();
121                         break;
122                     }else
123                         j++;
124                 }
125                 if(siguientes != null){
126                     j=0;
127                     while(j < siguientes.size()){
128                         String nombre = ""+(i+1);
129                         Object objeto = nombre;
130
131                         k=0;

```

```

131         while(k < this.gramatica.getTerm().size
132                ()){
133             if(siguientes.get(j).getNombre().
134                equals("$")){
135                 columna = this.matrizPred.
136                     getColumnCount()-1;
137                 break;
138             }else{
139                 if(this.gramatica.getTerm().get
140                    (k).getNombre().equals(
141                    siguientes.get(j).getNombre
142                    ())) {
143                     columna = k+1;
144                     break;
145                 }
146             }
147             k++;
148         }
149         setCeldaPredictiva(objeto, fila,
150                             columna);
151         j++;
152     }
153 }
154
155 public DefaultTableModel getTabla(){
156     return this.matrizPred;
157 }
158
159 public void setTabla(DefaultTableModel tabla){
160     this.matrizPred = tabla;
161 }
162
163 public Object getCeldaPredictiva(int i, int j){
164     return this.matrizPred.getValueAt(i, j);
165 }
166
167 public void setCeldaPredictiva(Object objeto, int i, int j){
168     this.matrizPred.setValueAt(objeto, i, j);
169 }
170
171 public void crearFunError(FuncionError fun){
172     this.funError.add(fun);
173 }
174
175 public ArrayList<FuncionError> getFunError() {
176

```

```

177         return funError;
178     }
179
180     public void setFunError(ArrayList<FuncionError> funcion){
181         this.funError = funcion;
182     }
183 }

```

## 2.4.20. VentanaCreacionGramatica.java

```

1  // SimAS / Editor
2  // Ventana Creacion Gramatica
3
4  package es.uco.simas.editor;
5
6  import es.uco.simas.util.gramatica.Gramatica;
7  import javax.swing.DefaultListModel;
8
9  /**
10   * @author vanesa
11   */
12  public class VentanaCreacionGramatica extends javax.swing.JFrame {
13
14      private PanelCreacionGramaticaPaso1 paso1 ;
15      private PanelCreacionGramaticaPaso2 paso2 ;
16      private PanelCreacionGramaticaPaso3 paso3 ;
17      private PanelCreacionGramaticaPaso4 paso4 ;
18      private Editor ventanaPadre ;
19
20      public VentanaCreacionGramatica(Editor ventanaPadre, Gramatica gr)
21      {
22          initComponents();
23          this.setResizable(false);
24          this.ventanaPadre =ventanaPadre;
25          PanelCreacionGramaticaPaso1 paso1 = new
26              PanelCreacionGramaticaPaso1(this);
27          this.paso1 =paso1;
28          PanelCreacionGramaticaPaso2 paso2 = new
29              PanelCreacionGramaticaPaso2(this);
30          this.paso2 =paso2;
31          PanelCreacionGramaticaPaso3 paso3 = new
32              PanelCreacionGramaticaPaso3(this);
33          this.paso3 =paso3;
34          PanelCreacionGramaticaPaso4 paso4 = new
35              PanelCreacionGramaticaPaso4(this);
36          this.paso4 =paso4;
37          this.getContentPane().removeAll();
38          this.setContentPane(this.paso1);
39          this.pack();

```

```

36     this.validate();
37     this.setTitle("Asistente de creacion de gramáticas. Paso 1 de
38         4");
39     this.setVisible(true);
40
41     if (gr!=null){
42         this.paso1.setNombre(gr.getNombre());
43         this.paso1.setDescripcion(gr.getDescripcion());
44         this.paso2.asignarListaSimbolosTerminales(gr.getTerminales
45             ());
46         this.paso2.asignarListaSimbolosNoTerminales(gr.
47             getNoTerminales());
48         this.paso3.asignarProducciones(gr.getProducciones());
49         this.paso4.setSimboloInicial(gr.getSimbInicial());
50     }
51
52     public void cambiarPaso( int paso) {
53         switch(paso){
54             case 1: {
55                 this.setContentPane(this.paso1);
56                 this.pack();
57                 this.setVisible(true);
58                 this.validate();
59                 this.setTitle("Asistente de creacion de gramáticas. Paso 1
60                     de 4");
61                 break;
62             }
63             case 2: {
64                 this.setContentPane(this.paso2);
65                 this.pack();
66                 this.setVisible(true);
67                 this.validate();
68                 this.setTitle("Asistente de creacion de gramáticas. Paso 2
69                     de 4");
70                 break;
71             }
72             case 3: {
73                 this.setContentPane(this.paso3);
74                 this.pack();
75                 this.setVisible(true);
76                 this.validate();
77                 this.setTitle("Asistente de creacion de gramáticas. Paso 3
78                     de 4");
79                 break;
80             }
81             case 4: {
82                 this.setContentPane(this.paso4);
83                 this.pack();
84                 if (this.paso2.getSimbolosNoTerminales()!= null) {
85                     this.paso4.setSimbolosNoTerminales(this.paso2.
86                         getSimbolosNoTerminales());
87                 }
88             }
89         }
90     }

```

```

82         this.setVisible(true);
83         this.validate();
84         this.setTitle("Asistente de creacion de gramáticas. Paso 4
            de 4");
85     }
86 }
87 }
88
89     public DefaultListModel getSimbolosTerminales( ) {
90         return this.paso2.getSimbolosTerminales();
91     }
92
93     public DefaultListModel getSimbolosNoTerminales( ) {
94         return this.paso2.getSimbolosNoTerminales();
95     }
96
97     public DefaultListModel getProducciones(){
98         return this.paso3.getProducciones();
99     }
100
101     public void setProducciones(DefaultListModel pr){
102         this.paso3.asignarProducciones(pr);
103     }
104
105     public void finalizarAsistente( ) {
106         Gramatica gramatica= new Gramatica(this.paso1.
            getNombreGramatica(), this.paso1.getDescripcionGramatica())
            ;
107         gramatica.setVocabulario(this.paso2.getSimbolosNoTerminales(),
            this.paso2.getSimbolosTerminales());
108         gramatica.setNoTerminales(this.paso2.getSimbolosNoTerminales())
            ;
109         gramatica.setTerminales(this.paso2.getSimbolosTerminales());
110         gramatica.setProducciones(this.paso3.getProducciones());
111         gramatica.setSimbInicial(this.paso4.getSimboloInicial());
112
113         Editor p1 = new Editor(gramatica);
114         this.ventanaPadre.dispose();
115         p1.setVisible(true);
116         p1.setLocationRelativeTo(null);
117         this.dispose();
118     }
119
120     /**
121      * This method is called from within the constructor to initialize
122      * the form.
123      * WARNING: Do NOT modify this code. The content of this method is
124      * always
125      * regenerated by the Form Editor.
126      */
127     // @SuppressWarnings("unchecked")
128     // <editor-fold defaultstate="collapsed" desc="Generated Code">
129     GEN-BEGIN: initComponents

```

```

127     private void initComponents() {
128
129         jLabel1 = new javax.swing.JLabel();
130
131         setDefaultCloseOperation(javax.swing.WindowConstants.
            DISPOSE_ON_CLOSE);
132
133         jLabel1.setText("Ventana");
134
135         javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
            getContentPane());
136         getContentPane().setLayout(layout);
137         layout.setHorizontalGroup(
138             layout.createParallelGroup(javax.swing.GroupLayout.
                Alignment.LEADING)
139             .addGroup(layout.createSequentialGroup()
140                 .addGap(185, 185, 185)
141                 .addComponent(jLabel1)
142                 .addGap(370, Short.MAX_VALUE))
143             );
144         layout.setVerticalGroup(
145             layout.createParallelGroup(javax.swing.GroupLayout.
                Alignment.LEADING)
146             .addGroup(layout.createSequentialGroup()
147                 .addGap(39, 39, 39)
148                 .addComponent(jLabel1)
149                 .addGap(358, Short.MAX_VALUE))
150             );
151
152         pack();
153     } // </editor-fold> //GEN-END: initComponents
154 /**/
155 /**
156  * @param args the command line arguments
157  */
158
159 // Variables declaration - do not modify //GEN-BEGIN: variables
160 private javax.swing.JLabel jLabel1;
161 // End of variables declaration //GEN-END: variables
162
163 }

```

### 2.4.21. VentanaProducciones.java

```

1 // SimAS / Editor
2 // Ventana Producciones
3
4 package es.uco.simas.editor;
5
6 import java.util.ArrayList;

```



```

7 import javax.swing.DefaultComboBoxModel;
8 import javax.swing.DefaultListModel;
9 import javax.swing.JOptionPane;
10 import javax.swing.*;
11
12 /**
13  * @author vanesa
14  */
15 public class VentanaProducciones extends javax.swing.JFrame {
16
17     private DefaultListModel modeloProducciones ;
18     private PanelCreacionGramaticaPaso3 panelPadre ;
19     private int idxProduccionSeleccionada ;
20
21     public VentanaProducciones(PanelCreacionGramaticaPaso3 ventanaPadre
22         , ListModel producciones) {
23
24         this.idxProduccionSeleccionada = -1;
25         DefaultListModel list = new DefaultListModel();
26         this.modeloProducciones = list;
27         int i=0;
28         while(true) {
29             if(i >= producciones.getSize()) {
30                 break;
31             }
32             if(i < producciones.getSize()) {
33                 this.modeloProducciones.add(i, producciones.getElementAt
34                     (i));
35                 i = i+1;
36             }
37         }
38
39         initComponents();
40         this.jList1.setModel(producciones);
41         this.panelPadre = ventanaPadre;
42         inicializarCombos();
43     }
44
45     /**
46      * This method is called from within the constructor to initialize
47      * the form.
48      * WARNING: Do NOT modify this code. The content of this method is
49      * always
50      * regenerated by the Form Editor.
51      */
52     @SuppressWarnings("unchecked")
53     // <editor-fold defaultstate="collapsed" desc="Generated Code">
54     GEN-BEGIN: initComponents
55     private void initComponents() {
56
57         jPanel1 = new javax.swing.JPanel();
58         jScrollPane1 = new javax.swing.JScrollPane();
59         jList1 = new javax.swing.JList();

```

```

55     jLabel1 = new javax.swing.JLabel();
56     jLabel2 = new javax.swing.JLabel();
57     jComboBoxAntecedente = new javax.swing.JComboBox();
58     jTextFieldConsecuente = new javax.swing.JTextField();
59     jLabel4 = new javax.swing.JLabel();
60     jLabel5 = new javax.swing.JLabel();
61     jButtonInsertar = new javax.swing.JButton();
62     jButtonEliminar = new javax.swing.JButton();
63     jButtonAceptar = new javax.swing.JButton();
64     jButtonCancelar = new javax.swing.JButton();
65     jButtonBorrar = new javax.swing.JButton();
66     jLabel6 = new javax.swing.JLabel();
67     jButtonModificar = new javax.swing.JButton();
68     jLabel3 = new javax.swing.JLabel();
69     jLabel7 = new javax.swing.JLabel();
70     jScrollPane2 = new javax.swing.JScrollPane();
71     jListTerm = new javax.swing.JList();
72     jScrollPane3 = new javax.swing.JScrollPane();
73     jListNTerm = new javax.swing.JList();
74     jButton1 = new javax.swing.JButton();
75
76     setDefaultCloseOperation(javax.swing.WindowConstants.
        DISPOSE_ON_CLOSE);
77     setTitle("Producciones de la Gramática");
78
79     jPanel1.setBackground(new java.awt.Color(233, 244, 244));
80
81     jScrollPane1.setViewportView(jList1);
82
83     jLabel1.setFont(new java.awt.Font("Ubuntu", 1, 18)); // NOI18N
84     jLabel1.setText("Nueva Produccion:");
85
86     jLabel2.setText("Antecedente");
87
88     jLabel4.setText("Consecuente");
89
90     jLabel5.setText("Añadir símbolo al consecuente:");
91
92     jButtonInsertar.setText("Insertar Produccion");
93     jButtonInsertar.addActionListener(new java.awt.event.
        ActionListener() {
94         public void actionPerformed(java.awt.event.ActionEvent evt)
95         {
96             jButtonInsertarActionPerformed(evt);
97         }
98     });
99
100    jButtonEliminar.setText("Eliminar Produccion");
101    jButtonEliminar.addActionListener(new java.awt.event.
        ActionListener() {
102        public void actionPerformed(java.awt.event.ActionEvent evt)

```

```

103         }
104     });
105
106     jButtonAceptar.setText("Aceptar");
107     jButtonAceptar.addActionListener(new java.awt.event.
108         ActionListener() {
109         public void actionPerformed(java.awt.event.ActionEvent evt)
110         {
111             jButtonAceptarActionPerformed(evt);
112         }
113     });
114
115     jButtonCancelar.setText("Cancelar");
116     jButtonCancelar.addActionListener(new java.awt.event.
117         ActionListener() {
118         public void actionPerformed(java.awt.event.ActionEvent evt)
119         {
120             jButtonCancelarActionPerformed(evt);
121         }
122     });
123
124     jButtonBorrar.setIcon(new javax.swing.ImageIcon(getClass().
125         getResource("/es/uco/simas/resources/flechaI.png"))); //
126     NOI18N
127     jButtonBorrar.setToolTipText("Borrar");
128     jButtonBorrar.addActionListener(new java.awt.event.
129         ActionListener() {
130         public void actionPerformed(java.awt.event.ActionEvent evt)
131         {
132             jButtonBorrarActionPerformed(evt);
133         }
134     });
135
136     jLabel6.setIcon(new javax.swing.ImageIcon(getClass().
137         getResource("/es/uco/simas/resources/flechaD.png"))); //
138     NOI18N
139
140     jButtonModificar.setText("Modificar Produccion");
141     jButtonModificar.addActionListener(new java.awt.event.
142         ActionListener() {
143         public void actionPerformed(java.awt.event.ActionEvent evt)
144         {
145             jButtonModificarActionPerformed(evt);
146         }
147     });
148
149     jLabel3.setText("No Terminales");
150
151     jLabel7.setText("Terminales");
152
153     jListTerm.addMouseListener(new java.awt.event.MouseAdapter() {
154         public void mousePressed(java.awt.event.MouseEvent evt) {
155             addNTerminales(evt);
156         }
157     });

```

```

144     }
145   });
146   jScrollPane2.setViewportViewView(jListTerm);
147
148   jListNTerm.addMouseListener(new java.awt.event.MouseAdapter() {
149       public void mousePressed(java.awt.event.MouseEvent evt) {
150           addTerminales(evt);
151       }
152   });
153   jScrollPane3.setViewportViewView(jListNTerm);
154
155   jButton1.setText("\u03b5");
156   jButton1.addActionListener(new java.awt.event.ActionListener()
157   {
158       public void actionPerformed(java.awt.event.ActionEvent evt)
159       {
160           jButton1ActionPerformed(evt);
161       }
162   });
163
164   javax.swing.GroupLayout jPanel1Layout = new javax.swing.
165       GroupLayout(jPanel1);
166   jPanel1.setLayout(jPanel1Layout);
167   jPanel1Layout.setHorizontalGroup(
168       jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
169           Alignment.LEADING)
170       .addGroup(jPanel1Layout.createSequentialGroup()
171           .add(jPanel1Layout.createParallelGroup(javax.swing.
172               GroupLayout.Alignment.LEADING)
173               .addGroup(jPanel1Layout.createSequentialGroup()
174                   .add(jScrollPane2, javax.swing.
175                       GroupLayout.PREFERRED_SIZE, 200, javax.
176                           swing.GroupLayout.PREFERRED_SIZE)
177                   .addPreferredGap(javax.swing.LayoutStyle.
178                       GroupLayout.Alignment.LEADING)
179                   .add(jScrollPane3, javax.swing.

```

```

swing . GroupLayout .
    PREFERRED.SIZE))
179 .addGroup(jPanel1Layout .
    createSequentialGroup ())
180 .addPreferredGap (javax . swing .
    LayoutStyle .
    ComponentPlacement .RELATED)
181 .addComponent (jButtonModificar
    )))
182 .addGroup (jPanel1Layout .createParallelGroup
    (javax . swing . GroupLayout . Alignment .
    TRAILING)
183 .addComponent (jButtonEliminar)
184 .addComponent (jScrollPane1 , javax . swing
    . GroupLayout .PREFERRED.SIZE, 508 ,
    javax . swing . GroupLayout .
    PREFERRED.SIZE)))
185 .addGroup (jPanel1Layout .createSequentialGroup ())
186 .addGap (89 , 89 , 89)
187 .addComponent (jLabel3)
188 .addGap (233 , 233 , 233)
189 .addComponent (jLabel7)))
190 .addGap (0 , 0 , Short .MAX_VALUE))
191 .addGroup (javax . swing . GroupLayout . Alignment . TRAILING ,
    jPanel1Layout .createSequentialGroup ())
192 .addGap (28 , 28 , 28)
193 .addGroup (jPanel1Layout .createParallelGroup (javax . swing
    . GroupLayout . Alignment .LEADING)
194 .addGroup (jPanel1Layout .createSequentialGroup ())
195 .addComponent (jLabel5)
196 .addGap (40 , 40 , 40)
197 .addComponent (jButton1 , javax . swing . GroupLayout
    .PREFERRED.SIZE, 89 , javax . swing .
    GroupLayout .PREFERRED.SIZE))
198 .addGroup (jPanel1Layout .createSequentialGroup ())
199 .addGroup (jPanel1Layout .createParallelGroup (
    javax . swing . GroupLayout . Alignment .LEADING)
200 .addGroup (jPanel1Layout .
    createSequentialGroup ())
201 .addComponent (jLabel2)
202 .addGap (55 , 55 , 55))
203 .addComponent (jComboBoxAntecedente , javax .
    swing . GroupLayout . Alignment . TRAILING ,
    javax . swing . GroupLayout .PREFERRED.SIZE,
    151 , javax . swing . GroupLayout .
    PREFERRED.SIZE))
204 .addGroup (jPanel1Layout .createParallelGroup (
    javax . swing . GroupLayout . Alignment .LEADING)
205 .addGroup (jPanel1Layout .
    createSequentialGroup ())
206 .addGap (93 , 93 , 93)
207 .addComponent (jLabel4))

```

```

208         .addGroup(jPanel1Layout .
209             createSequentialGroup ()
210             .addGap(18, 18, 18)
211             .addComponent(jLabel6)
212             .addPreferredGap(javax.swing .
213                 LayoutStyle .ComponentPlacement .
214                     RELATED)
215             .addComponent(jTextFieldConsecuente ,
216                 javax.swing .GroupLayout .
217                     PREFERRED_SIZE, 274, javax.swing .
218                         GroupLayout .PREFERRED_SIZE)
219             .addPreferredGap(javax.swing .
220                 LayoutStyle .ComponentPlacement .
221                     RELATED)
222             .addComponent(jButtonBorrar , javax .
223                 swing .GroupLayout .PREFERRED_SIZE,
224                 37, javax .swing .GroupLayout .
225                     PREFERRED_SIZE)))
226     .addGroup(jPanel1Layout .createSequentialGroup ()
227         .addGap(191, 191, 191)
228         .addComponent(jLabel1))
229     .addGap(169, 169, 169))
230 .addGroup(javax.swing .GroupLayout .Alignment .TRAILING,
231     jPanel1Layout .createSequentialGroup ()
232     .addGap(64, 64, 64)
233     .addComponent(jButtonCancelar)
234     .addGap(248, 248, 248)
235     .addComponent(jButtonAceptar)
236     .addContainerGap(javax.swing .GroupLayout .DEFAULT_SIZE,
237         Short .MAX_VALUE))
238 );
239 jPanel1Layout .setVerticalGroup(
240     jPanel1Layout .createParallelGroup(javax.swing .GroupLayout .
241         Alignment .LEADING)
242     .addGroup(javax.swing .GroupLayout .Alignment .TRAILING,
243         jPanel1Layout .createSequentialGroup ()
244         .addContainerGap ()
245         .addComponent(jLabel1)
246         .addGap(14, 14, 14)
247         .addGroup(jPanel1Layout .createParallelGroup(javax.swing
248             .GroupLayout .Alignment .BASELINE)
249             .addComponent(jLabel2)
250             .addComponent(jLabel4))
251         .addPreferredGap(javax.swing .LayoutStyle .
252             ComponentPlacement .UNRELATED)
253         .addGroup(jPanel1Layout .createParallelGroup(javax.swing
254             .GroupLayout .Alignment .LEADING)
255             .addComponent(jLabel6 , javax.swing .GroupLayout .
256                 Alignment .TRAILING, javax.swing .GroupLayout .
257                     PREFERRED_SIZE, 27, javax.swing .GroupLayout .
258                         PREFERRED_SIZE)
259             .addGroup(jPanel1Layout .createParallelGroup(javax .
260                 swing .GroupLayout .Alignment .BASELINE)

```

```

239         .addComponent(jComboBoxAntecedente, javax.swing
        . GroupLayout.PREFERRED_SIZE, javax.swing.
        GroupLayout.DEFAULT_SIZE, javax.swing.
        GroupLayout.PREFERRED_SIZE)
240         .addComponent(jTextFieldConsecuente, javax.
        swing. GroupLayout.PREFERRED_SIZE, javax.
        swing. GroupLayout.DEFAULT_SIZE, javax.swing
        . GroupLayout.PREFERRED_SIZE))
241         .addComponent(jButtonBorrar))
242     .addPreferredGap(javax.swing.LayoutStyle.
        ComponentPlacement.UNRELATED)
243     .addGroup(jPanel1Layout.createParallelGroup(javax.swing
        . GroupLayout.Alignment.LEADING)
244         .addComponent(jLabel5)
245         .addComponent(jButton1))
246     .addPreferredGap(javax.swing.LayoutStyle.
        ComponentPlacement.RELATED, javax.swing. GroupLayout
        .DEFAULT_SIZE, Short.MAX_VALUE)
247     .addGroup(jPanel1Layout.createParallelGroup(javax.swing
        . GroupLayout.Alignment.BASELINE)
248         .addComponent(jLabel3)
249         .addComponent(jLabel7))
250     .addPreferredGap(javax.swing.LayoutStyle.
        ComponentPlacement.RELATED)
251     .addGroup(jPanel1Layout.createParallelGroup(javax.swing
        . GroupLayout.Alignment.LEADING, false)
252         .addComponent(jScrollPane3)
253         .addComponent(jScrollPane2))
254     .addGap(27, 27, 27)
255     .addGroup(jPanel1Layout.createParallelGroup(javax.swing
        . GroupLayout.Alignment.BASELINE)
256         .addComponent(jButtonInsertar)
257         .addComponent(jButtonModificar)
258         .addComponent(jButtonEliminar))
259     .addPreferredGap(javax.swing.LayoutStyle.
        ComponentPlacement.RELATED)
260     .addComponent(jScrollPane1, javax.swing. GroupLayout.
        PREFERRED_SIZE, 148, javax.swing. GroupLayout.
        PREFERRED_SIZE)
261     .addGap(23, 23, 23)
262     .addGroup(jPanel1Layout.createParallelGroup(javax.swing
        . GroupLayout.Alignment.BASELINE)
263         .addComponent(jButtonCancelar)
264         .addComponent(jButtonAceptar))
265     .addGap(77, 77, 77))
266 );
267
268 javax.swing. GroupLayout layout = new javax.swing. GroupLayout(
        getContentPane());
269 getContentPane().setLayout(layout);
270 layout.setHorizontalGroup(
271     layout.createParallelGroup(javax.swing. GroupLayout.
        Alignment.LEADING)

```

```

272         .addComponent(jPanel1 , javax.swing.GroupLayout .
                PREFERRED_SIZE, 584, javax.swing.GroupLayout .
                PREFERRED_SIZE)
273     );
274     layout.setVerticalGroup(
275         layout.createParallelGroup(javax.swing.GroupLayout .
                Alignment.LEADING)
276         .addComponent(jPanel1 , javax.swing.GroupLayout .
                PREFERRED_SIZE, 584, javax.swing.GroupLayout .
                PREFERRED_SIZE)
277     );
278
279     pack();
280 }// </editor-fold>//GEN-END: initComponents
281
282 private void jButtonBorrarActionPerformed(java.awt.event .
    ActionEvent evt) { //GEN-FIRST:
    event_jButtonBorrarActionPerformed
283     Boolean espacioEncontrado= null;
284     String cadenaConsecuente= null;
285     String espacioBlanco= null;
286     int indice= 0;
287     cadenaConsecuente=this.jTextFieldConsecuente.getText();
288     indice=0;
289     espacioEncontrado=Boolean.valueOf(false);
290     espacioBlanco="";
291
292     if(cadenaConsecuente.length()!= 0){
293         if(cadenaConsecuente.charAt((cadenaConsecuente.length()- 1)
                ) == 32) {
294             cadenaConsecuente =cadenaConsecuente.substring(0,(
                cadenaConsecuente.length()- 2));
295
296         }
297         if(cadenaConsecuente.length() > 0) {
298             indice=(cadenaConsecuente.length()- 1);
299             while(true) {
300                 if(indice<=0) {
301                     break;
302                 }
303                 if(indice > 0) {
304                     if(String.valueOf(cadenaConsecuente.charAt(
                        indice)).equals(" ")!=false) {
305                         espacioEncontrado =Boolean.valueOf(true);
306                         break ;
307                     }
308                 }
309                 indice = indice+(-1);
310             }
311         }
312     }
313     if(espacioEncontrado.booleanValue()) {

```



```

314         this.jTextFieldConsecuente.setText(cadenaConsecuente.
315             substring(0, indice));
316     } else {
317         this.jTextFieldConsecuente.setText("");
318     }
319 }
320 }//GEN-LAST:event_jButtonBorrarActionPerformed
321
322 private void jButtonCancelarActionPerformed(java.awt.event.
323     ActionEvent evt) { //GEN-FIRST:
324     event_jButtonCancelarActionPerformed
325     int conf = JOptionPane.showConfirmDialog(null, "¿Desea salir
326     de la edicion de las producciones?", "Salir", JOptionPane.
327     YES.NO.OPTION);
328
329     if (conf==0)
330     this.dispose();
331 }//GEN-LAST:event_jButtonCancelarActionPerformed
332
333 private void jButtonAceptarActionPerformed(java.awt.event.
334     ActionEvent evt) { //GEN-FIRST:
335     event_jButtonAceptarActionPerformed
336     if (this.panelPadre!= null) {
337         if (this.jList1.getModel().getSize() > 0) {
338             this.panelPadre.asignarProducciones((DefaultListModel)
339                 this.jList1.getModel());
340         }
341     }
342     this.dispose();
343 }//GEN-LAST:event_jButtonAceptarActionPerformed
344
345 private void jButtonEliminarActionPerformed(java.awt.event.
346     ActionEvent evt) { //GEN-FIRST:
347     event_jButtonEliminarActionPerformed
348     DefaultListModel modelo= null;
349     int seleccion= 0;
350     seleccion=this.jList1.getSelectedIndex();
351     if (seleccion != -1) {
352         modelo=(DefaultListModel) this.jList1.getModel();
353         modelo.remove(seleccion);
354     }
355 }//GEN-LAST:event_jButtonEliminarActionPerformed
356
357 private void jButtonInsertarActionPerformed(java.awt.event.
358     ActionEvent evt) { //GEN-FIRST:
359     event_jButtonInsertarActionPerformed
360     String produccion= null;
361     int encontrado= 0;
362     int i= 0;
363
364     if (this.jButtonInsertar.getText().equals("Insertar Produccion"))
365     {

```

```

354         if (this.jTextFieldConsecuente.getText().length() > 0) {
355             encontrado=0;
356             i=0;
357             while(true) {
358                 if(i >= this.modeloProducciones.getSize()){
359                     break;
360                 }
361                 if(i < this.modeloProducciones.getSize()) {
362                     produccion=this.modeloProducciones.getElementAt
                        (i).toString();
363                     StringBuilder aux = new StringBuilder();
364                     if(produccion.equals(aux.append(this.
                        jComboBoxAntecedente.getSelectedItemAt
                        toString()).append(" ").append("\u2192").
                        append(" ").append(this.
                        jTextFieldConsecuente.getText()).append(" "
                        ).toString())!=false) {
365                         encontrado=1;
366                     }
367                     i = i+1;
368                 }
369             }
370             if(encontrado==0) {
371                 StringBuilder prod = new StringBuilder();
372                 this.modeloProducciones.add(this.modeloProducciones
                        .getSize(),prod.append(this.
                        jComboBoxAntecedente.getSelectedItemAt
                        toString()).append("\u2192").append(" ").append(this.
                        jTextFieldConsecuente.getText()).append(" ").
                        toString());
373                 this.jList1.setModel(this.modeloProducciones);
374                 this.jTextFieldConsecuente.requestFocus();
375                 this.jTextFieldConsecuente.setText("");
376             }
377         }
378     } else{
379         StringBuilder prod = new StringBuilder();
380         this.modeloProducciones.setElementAt(prod.append(this.
            jComboBoxAntecedente.getSelectedItemAt
            toString()).append("\u2192").append(" ").append(this.
            jTextFieldConsecuente.getText()).append(" ").toString()
            , this.jList1.getSelectedIndex()); //add(this.
            modeloProducciones.getSize(),prod.append(this.
            jComboBoxAntecedente.getSelectedItemAt
            toString()).append("\u2192").append(" ").append(this.
            jTextFieldConsecuente.getText()).append(" ").toString()
            );
381         this.jList1.setModel(this.modeloProducciones);
382         this.jTextFieldConsecuente.requestFocus();
383         this.jTextFieldConsecuente.setText("");
384         this.jButtonInsertar.setText("Insertar Produccion");
385     }
386 } //GEN-LAST:event_jButtonInsertarActionPerformed

```

```

387
388     private void jButtonModificarActionPerformed(java.awt.event.
        ActionEvent evt) {//GEN-FIRST: event_jButtonModificarActionPerformed
389         String produccion= null;
390         int i= 0;
391
392         if( this.jList1.getSelectedIndex()>=0) {
393             produccion=this.jList1.getSelectedValue().toString();
394             this.idxProduccionSeleccionada = this.jList1.
                getSelectedIndex();
395
396             i=0;
397             while(true) {
398                 if(i >= this.jComboBoxAntecedente.getItemCount()) {
399                     break;
400                 }
401                 if(i < this.jComboBoxAntecedente.getItemCount()) {
402                     if( this.jComboBoxAntecedente.getItemAt(i).toString
                        ().equals(produccion.substring(0,1))!=false){
403                         this.jComboBoxAntecedente.setSelectedIndex(i);
404                     }
405                     i = i+1;
406                 }
407             }
408             this.jTextFieldConsecuente.setText(produccion.substring(4))
                ;
409             this.jButtonInsertar.setText("Guardar cambios");
410         }
411     }//GEN-LAST: event_jButtonModificarActionPerformed
412
413     private void addTerminales(java.awt.event.MouseEvent evt) {//GEN-
        FIRST: event_addTerminales
414         if( this.jTextFieldConsecuente.getText().length()==0) {
415             StringBuilder string = new StringBuilder();
416             this.jTextFieldConsecuente.setText(string.append(this.
                jTextFieldConsecuente.getText()).append(this.jListNTerm
                .getSelectedValue()).toString());
417         } else {
418             //Si se ha insertado epsilon en el consecuente no se puede
                insertar ningun simbolo
419             int i=0;
420             int j=0;
421             String [] aux = this.jTextFieldConsecuente.getText().split(
                " ");
422
423             while(i < aux.length){
424                 if("\u03b5".equals(aux[i])){
425                     JOptionPane.showConfirmDialog(null, "El símbolo \
                        u03b5 debe aparecer solo en el consecuente de
                        una produccion.", "Error",JOptionPane.
                        CLOSED_OPTION);
426                     j=1;

```

```

427         break;
428     }
429     i++;
430 }
431 //Si el consecuente no tiene epsilon se inserta el simbolo
seleccionado
432 if(j==0){
433     if(this.jTextFieldConsecuente.getText().charAt((this.
jTextFieldConsecuente.getText().length()- 1)) !=
32){
434         StringBuilder string2 = new StringBuilder();
435         this.jTextFieldConsecuente.setText(string2.append(
this.jTextFieldConsecuente.getText()).append("
").append(this.jListNTerm.getSelectedValue()).
toString());
436     } else {
437         StringBuilder string3 = new StringBuilder();
438         this.jTextFieldConsecuente.setText(string3.append(
this.jTextFieldConsecuente.getText()).append(
this.jListNTerm.getSelectedValue()).toString())
;
439     }
440 }
441 }
442 //GEN-LAST: event_addTerminales
443
444 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt
) //GEN-FIRST: event_jButton1ActionPerformed
445     if(this.jTextFieldConsecuente.getText().length() == 0) {
446         StringBuilder string = new StringBuilder();
447         this.jTextFieldConsecuente.setText(string.append(this.
jTextFieldConsecuente.getText()).append("\u03b5").
toString());
448     } else {
449         JOptionPane.showConfirmDialog(null, "El símbolo \u03b5
debe aparecer solo en el consecuente de una produccion.
", "Error", JOptionPane.CLOSED_OPTION);
450     }
451 //GEN-LAST: event_jButton1ActionPerformed
452
453 private void addNTerminales(java.awt.event.MouseEvent evt) //GEN-
FIRST: event_addNTerminales
454     if(this.jTextFieldConsecuente.getText().length()==0) {
455         StringBuilder string = new StringBuilder();
456         this.jTextFieldConsecuente.setText(string.append(this.
jTextFieldConsecuente.getText()).append(this.jListTerm.
getSelectedValue()).toString());
457     } else {
458         //Si se ha insertado epsilon en el consecuente no se puede
insertar ningun simbolo
459         int i=0;
460         int j=0;

```

```

461         String [] aux = this.jTextFieldConsecuente.getText().split(
462             " ");
463         while(i < aux.length){
464             if("\\u03b5".equals(aux[i])){
465                 JOptionPane.showConfirmDialog(null, "El símbolo \\
466                 u03b5 debe aparecer solo en el consecuente de
                 una produccion.", "Error",JOptionPane.
                 CLOSED_OPTION);
467                 j=1;
468                 break;
469             }
470             i++;
471         }
472         //Si el consecuente no tiene epsilon se inserta el simbolo
                 seleccionado
473         if(j==0){
474             if(this.jTextFieldConsecuente.getText().charAt((this.
                 jTextFieldConsecuente.getText().length()- 1)) !=
                 32){
475                 StringBuilder string2 = new StringBuilder();
476                 this.jTextFieldConsecuente.setText(string2.append(
                 this.jTextFieldConsecuente.getText()).append("
                 ").append(this.jListTerm.getSelectedValue()).
                 toString());
477             } else {
478                 StringBuilder string3 = new StringBuilder();
479                 this.jTextFieldConsecuente.setText(string3.append(
480                 this.jTextFieldConsecuente.getText()).append(
                 this.jListTerm.getSelectedValue()).toString());
481             }
482         }
483     }
484     } //GEN-LAST: event_addNTerminales
485
486     private void inicializarCombos( ) {
487         ArrayList<String> listaAntecedente= null;
488         ArrayList<String> listaConsecuente= null;
489         DefaultComboBoxModel modeloAntecedente= null;
490         DefaultComboBoxModel modeloConsecuente= null;
491         DefaultListModel noTerminales= null;
492         DefaultListModel terminales= null;
493         int i= 0;
494         terminales=this.panelPadre.getSimbolosTerminales();
495         noTerminales=this.panelPadre.getSimbolosNoTerminales();
496         ArrayList antecedente = new ArrayList();
497         listaAntecedente=antecedente;
498         ArrayList consecuente = new ArrayList();
499         listaConsecuente=consecuente;
500
501         while(true) {

```

```

502         if(i >= noTerminales.size()) {
503             break;
504         }
505         if(i < noTerminales.size()) {
506             listaAntecedente.add((String)noTerminales.get(i));
507             i = i+1;
508         }
509     }
510     i=0;
511     while(true) {
512         if(i >= noTerminales.size()) {
513             break;
514         }
515         if(i < noTerminales.size()) {
516             listaConsecuente.add((String)noTerminales.get(i));
517             i = i+1;
518         }
519     }
520     i=0;
521     while(true) {
522         if(i >= terminales.size()) {
523             break;
524         }
525         if(i < terminales.size()) {
526             listaConsecuente.add((String)terminales.get(i));
527             i = i+1;
528         }
529     }
530     DefaultComboBoxModel comboAntecedente = new
        DefaultComboBoxModel(listaAntecedente.toArray());
531     modeloAntecedente=comboAntecedente;
532     DefaultComboBoxModel comboConsecuente = new
        DefaultComboBoxModel(listaConsecuente.toArray());
533     modeloConsecuente=comboConsecuente;
534     this.jComboBoxAntecedente.setModel(modeloAntecedente);
535     this.jListNTerm.setModel(terminales);
536     this.jListTerm.setModel(noTerminales);
537 }
538
539 /**
540  * @param args the command line arguments
541  */
542 public static void main(String args[]) {
543     /* Set the Nimbus look and feel */
544     //<editor-fold defaultstate="collapsed" desc=" Look and feel
        setting code (optional) ">
545     /* If Nimbus (introduced in Java SE 6) is not available, stay
        with the default look and feel.
546     * For details see http://download.oracle.com/javase/tutorial/
        uiswing/lookandfeel/plaf.html
547     */
548     try {

```

```

549         for (javax.swing.UIManager.LookAndFeelInfo info : javax.
swing.UIManager.getInstalledLookAndFeels()) {
550             if ("Nimbus".equals(info.getName())) {
551                 javax.swing.UIManager.setLookAndFeel(info.
getClassName());
552                 break;
553             }
554         }
555     } catch (ClassNotFoundException ex) {
556         java.util.logging.Logger.getLogger(VentanaProducciones.
class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
557     } catch (InstantiationException ex) {
558         java.util.logging.Logger.getLogger(VentanaProducciones.
class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
559     } catch (IllegalAccessException ex) {
560         java.util.logging.Logger.getLogger(VentanaProducciones.
class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
561     } catch (javax.swing.UnsupportedLookAndFeelException ex) {
562         java.util.logging.Logger.getLogger(VentanaProducciones.
class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
563     }
564     //</editor-fold>
565
566     /* Create and display the form */
567     java.awt.EventQueue.invokeLater(new Runnable() {
568         public void run() {
569             // new VentanaProducciones().setVisible(true);
570         }
571     });
572 }
573 // Variables declaration - do not modify//GEN-BEGIN:variables
574 private javax.swing.JButton jButton1;
575 private javax.swing.JButton jButtonAceptar;
576 private javax.swing.JButton jButtonBorrar;
577 private javax.swing.JButton jButtonCancelar;
578 private javax.swing.JButton jButtonEliminar;
579 private javax.swing.JButton jButtonInsertar;
580 private javax.swing.JButton jButtonModificar;
581 private javax.swing.JComboBox jComboBoxAntecedente;
582 private javax.swing.JLabel jLabel1;
583 private javax.swing.JLabel jLabel2;
584 private javax.swing.JLabel jLabel3;
585 private javax.swing.JLabel jLabel4;
586 private javax.swing.JLabel jLabel5;
587 private javax.swing.JLabel jLabel6;
588 private javax.swing.JLabel jLabel7;
589 private javax.swing.JList jList1;
590 private javax.swing.JList jListNTerm;
591 private javax.swing.JList jListTerm;

```

```

592     private javax.swing.JPanel jPanel1;
593     private javax.swing.JScrollPane jScrollPane1;
594     private javax.swing.JScrollPane jScrollPane2;
595     private javax.swing.JScrollPane jScrollPane3;
596     private javax.swing.JTextField jTextFieldConsecuente;
597     // End of variables declaration//GEN-END:variables
598 }

```

### 2.4.22. VentanaResultadosValidacion.java

```

1  //SimAS / Editor
2  //Ventana Resultados Validacion
3
4  package es.uco.simas.editor;
5
6  import java.util.ArrayList;
7
8  /**
9   * @author vanesa
10  */
11 public class VentanaResultadosValidacion extends javax.swing.JFrame {
12
13     private int estadoValidacion ;
14     private ArrayList<String> mensajesError ;
15
16     public VentanaResultadosValidacion(int estadoValidacion , ArrayList
17     <String> mensajesError){
18         this.estadoValidacion =estadoValidacion;
19         this.mensajesError =mensajesError;
20         initComponents();
21     }
22
23     public VentanaResultadosValidacion() {
24         initComponents();
25     }
26
27     /**
28      * This method is called from within the constructor to initialize
29      * the form.
30      * WARNING: Do NOT modify this code. The content of this method is
31      * always
32      * regenerated by the Form Editor.
33      */
34     @SuppressWarnings("unchecked")
35     // <editor-fold defaultstate="collapsed" desc="Generated Code">
36     GEN-BEGIN: initComponents
37     private void initComponents() {
38
39         jPanel1 = new javax.swing.JPanel();

```



```

37     setDefaultCloseOperation(javax.swing.WindowConstants.
        EXIT_ON_CLOSE);
38     setBackground(new java.awt.Color(233, 244, 244));
39
40     jPanel1.setBackground(new java.awt.Color(233, 244, 244));
41     jPanel1.setToolTipText("Validacion");
42
43     javax.swing.GroupLayout jPanel1Layout = new javax.swing.
        GroupLayout(jPanel1);
44     jPanel1.setLayout(jPanel1Layout);
45     jPanel1Layout.setHorizontalGroup(
46         jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
            Alignment.LEADING)
47         .addGap(0, 437, Short.MAX_VALUE)
48     );
49     jPanel1Layout.setVerticalGroup(
50         jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
            Alignment.LEADING)
51         .addGap(0, 323, Short.MAX_VALUE)
52     );
53
54     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
        getContentPane());
55     getContentPane().setLayout(layout);
56     layout.setHorizontalGroup(
57         layout.createParallelGroup(javax.swing.GroupLayout.
            Alignment.LEADING)
58         .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE
            , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE
            )
59     );
60     layout.setVerticalGroup(
61         layout.createParallelGroup(javax.swing.GroupLayout.
            Alignment.LEADING)
62         .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE
            , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE
            )
63     );
64
65     pack();
66 } // </editor-fold> // GEN-END: initComponents
67
68 /**
69  * @param args the command line arguments
70  */
71 public static void main(String args[]) {
72     /* Set the Nimbus look and feel */
73     //<editor-fold defaultstate="collapsed" desc="Look and feel
        setting code (optional) ">
74     /* If Nimbus (introduced in Java SE 6) is not available, stay
        with the default look and feel.
75     * For details see http://download.oracle.com/javase/tutorial/
        uiswing/lookandfeel/plaf.html

```

```

76      */
77      try {
78          for (javax.swing.UIManager.LookAndFeelInfo info : javax.
              swing.UIManager.getInstalledLookAndFeels()) {
79              if ("Nimbus".equals(info.getName())) {
80                  javax.swing.UIManager.setLookAndFeel(info.
                      getClassName());
81                  break;
82              }
83          }
84      } catch (ClassNotFoundException ex) {
85          java.util.logging.Logger.getLogger(
              VentanaResultadosValidacion.class.getName()).log(java.
              util.logging.Level.SEVERE, null, ex);
86      } catch (InstantiationException ex) {
87          java.util.logging.Logger.getLogger(
              VentanaResultadosValidacion.class.getName()).log(java.
              util.logging.Level.SEVERE, null, ex);
88      } catch (IllegalAccessException ex) {
89          java.util.logging.Logger.getLogger(
              VentanaResultadosValidacion.class.getName()).log(java.
              util.logging.Level.SEVERE, null, ex);
90      } catch (javax.swing.UnsupportedLookAndFeelException ex) {
91          java.util.logging.Logger.getLogger(
              VentanaResultadosValidacion.class.getName()).log(java.
              util.logging.Level.SEVERE, null, ex);
92      }
93      //</editor-fold>
94
95      /* Create and display the form */
96      java.awt.EventQueue.invokeLater(new Runnable() {
97          public void run() {
98              new VentanaResultadosValidacion().setVisible(true);
99          }
100     });
101 }
102 // Variables declaration - do not modify//GEN-BEGIN:variables
103 private javax.swing.JPanel jPanel1;
104 // End of variables declaration//GEN-END:variables
105 }

```

### 2.4.23. VentanaSimbolosNoTerminales.java

```

1 // SimAS / Editor
2 // Ventana Simbolos No Terminales
3
4 package es.uco.simas.editor;
5
6 import es.uco.simas.util.gramatica.Gramatica;
7 import java.util.ArrayList;

```

```

8 import javax.swing.JOptionPane;
9 import javax.swing.DefaultListModel;
10 import javax.swing.ListModel;
11
12 /**
13  * @author vanesa
14  */
15 public class VentanaSimbolosNoTerminales extends javax.swing.JFrame {
16
17     private DefaultListModel modeloListaTerminales ;
18     private DefaultListModel modeloListaNoTerminales ;
19     private PanelCreacionGramaticaPaso2 panelPadre ;
20     private Gramatica gramatica;
21
22     public VentanaSimbolosNoTerminales(ListModel simbolosNoTerminales ,
23         ListModel simbolosTerminales , PanelCreacionGramaticaPaso2
24         ventanaPadre , Gramatica gr) {
25
26         int i=0;
27         DefaultListModel NoTerminales = new DefaultListModel();
28         this.modeloListaNoTerminales =NoTerminales;
29
30         while(true) {
31             if(i >= simbolosNoTerminales.getSize()) {
32                 break;
33             }
34             if(i < simbolosNoTerminales.getSize()) {
35                 this.modeloListaNoTerminales.add(i,simbolosNoTerminales
36                     .getElementAt(i));
37                 i = i+1;
38             }
39         }
40
41         initComponents();
42         this.jList1.setModel(simbolosNoTerminales);
43         this.setResizable(false);
44         this.panelPadre =ventanaPadre;
45         this.gramatica = gr;
46         this.jTextField1.setText("");
47     }
48
49     /**
50      * This method is called from within the constructor to initialize
51      * the form.
52      * WARNING: Do NOT modify this code. The content of this method is
53      * always
54      * regenerated by the Form Editor.
55      */
56     @SuppressWarnings("unchecked")
57     // <editor-fold defaultstate="collapsed" desc="Generated Code">
58     GEN-BEGIN: initComponents
59     private void initComponents() {

```

```

55     jPanel1 = new javax.swing.JPanel();
56     jButtonAceptar = new javax.swing.JButton();
57     jButtonCancelar = new javax.swing.JButton();
58     jTextField1 = new javax.swing.JTextField();
59     jButtonInsertar = new javax.swing.JButton();
60     jScrollPane2 = new javax.swing.JScrollPane();
61     jList1 = new javax.swing.JList();
62     jButtonEliminar = new javax.swing.JButton();
63
64     setDefaultCloseOperation(javax.swing.WindowConstants.
        DISPOSE_ON_CLOSE);
65     setTitle("Símbolos No Terminales");
66
67     jPanel1.setBackground(new java.awt.Color(233, 244, 244));
68
69     jButtonAceptar.setText("Aceptar");
70     jButtonAceptar.addActionListener(new java.awt.event.
        ActionListener() {
71         public void actionPerformed(java.awt.event.ActionEvent evt)
72         {
73             jButtonAceptarActionPerformed(evt);
74         }
75     });
76
77     jButtonCancelar.setText("Cancelar");
78     jButtonCancelar.addActionListener(new java.awt.event.
        ActionListener() {
79         public void actionPerformed(java.awt.event.ActionEvent evt)
80         {
81             jButtonCancelarActionPerformed(evt);
82         }
83     });
84
85     jButtonInsertar.setText("Insertar");
86     jButtonInsertar.addActionListener(new java.awt.event.
        ActionListener() {
87         public void actionPerformed(java.awt.event.ActionEvent evt)
88         {
89             jButtonInsertarActionPerformed(evt);
90         }
91     });
92
93     jList1.setSelectionMode(javax.swing.ListSelectionModel.
        SINGLE_SELECTION);
94     jScrollPane2.setViewportView(jList1);
95
96     jButtonEliminar.setText("Eliminar");
97     jButtonEliminar.addActionListener(new java.awt.event.
        ActionListener() {
98         public void actionPerformed(java.awt.event.ActionEvent evt)
99         {
100             jButtonEliminarActionPerformed(evt);
101         }
102     });

```

```
98     });
99
100    javax.swing.GroupLayout jPanel1Layout = new javax.swing.
        GroupLayout(jPanel1);
101    jPanel1.setLayout(jPanel1Layout);
102    jPanel1Layout.setHorizontalGroup(
103        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
            Alignment.LEADING)
104        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            jPanel1Layout.createSequentialGroup()
105            .addGap(javax.swing.GroupLayout.DEFAULT_SIZE,
                Short.MAX_VALUE)
106            .addComponent(jButtonCancelar)
107            .addGap(18, 18, 18)
108            .addComponent(jButtonAceptar)
109            .addGap(javax.swing.GroupLayout.DEFAULT_SIZE,
                Short.MAX_VALUE)
110            .addGroup(jPanel1Layout.createSequentialGroup()
111                .addGap(34, 34, 34)
112                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.
                    GroupLayout.Alignment.LEADING, false)
113                    .addComponent(jScrollPane2)
114                    .addGroup(jPanel1Layout.createSequentialGroup()
115                        .addComponent(jTextField1, javax.swing.
                            GroupLayout.PREFERRED_SIZE, 287, javax.
                                swing.GroupLayout.PREFERRED_SIZE)
116                        .addGap(26, 26, 26)
117                        .addComponent(jButtonInsertar)))
118                .addPreferredGap(javax.swing.LayoutStyle.
                    ComponentPlacement.RELATED)
119                .addComponent(jButtonEliminar)
120                .addGap(javax.swing.GroupLayout.DEFAULT_SIZE,
                    Short.MAX_VALUE))
121        );
122    jPanel1Layout.setVerticalGroup(
123        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
            Alignment.LEADING)
124        .addGroup(jPanel1Layout.createSequentialGroup()
125            .addGap(40, 40, 40)
126            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.
                GroupLayout.Alignment.BASELINE)
127                .addComponent(jTextField1, javax.swing.GroupLayout.
                    PREFERRED_SIZE, javax.swing.GroupLayout.
                        DEFAULT_SIZE, javax.swing.GroupLayout.
                            PREFERRED_SIZE)
128                .addComponent(jButtonInsertar)
129                .addComponent(jButtonEliminar))
130            .addPreferredGap(javax.swing.LayoutStyle.
                ComponentPlacement.RELATED)
131            .addComponent(jScrollPane2, javax.swing.GroupLayout.
                DEFAULT_SIZE, 251, Short.MAX_VALUE)
132            .addGap(18, 18, 18)
133            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.
                GroupLayout.Alignment.BASELINE)
134                .addComponent(jButtonAceptar)
```

```

135         .addComponent(jButtonCancelar))
136         .addContainerGap()
137     );
138
139     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
140         getContentPane());
141     getContentPane().setLayout(layout);
142     layout.setHorizontalGroup(
143         layout.createParallelGroup(javax.swing.GroupLayout.
144             Alignment.LEADING)
145             .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE
146                 , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE
147             )
148     );
149     layout.setVerticalGroup(
150         layout.createParallelGroup(javax.swing.GroupLayout.
151             Alignment.LEADING)
152             .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE
153                 , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE
154             )
155     );
156     pack();
157 } // </editor-fold> //GEN-END: initComponents
158
159 private void jButtonCancelarActionPerformed(java.awt.event.
160     ActionEvent evt) { //GEN-FIRST:
161     event_jButtonCancelarActionPerformed
162     int conf = JOptionPane.showConfirmDialog(null, "¿Desea salir
163         de la edicion de la gramática?", "Salir",JOptionPane.
164         YES.NO.OPTION);
165
166     if (conf==0){
167         this.dispose();
168     }
169 } //GEN-LAST: event_jButtonCancelarActionPerformed
170
171 private void jButtonInsertarActionPerformed(java.awt.event.
172     ActionEvent evt) { //GEN-FIRST:
173     event_jButtonInsertarActionPerformed
174     String elemento = jTextField1.getText();
175     int encontrado= 0;
176     int i= 0;
177
178     if(elemento.equals("")){
179         encontrado =1;
180     }else{
181         while(true){
182             if(i >= this.modeloListaNoTerminales.getSize()) {
183                 break;
184             }
185             if(i < this.modeloListaNoTerminales.getSize()) {

```

```

174         if (this.modeloListaNoTerminales.getElementAt(i).
175             equals(elemento)) {
176             encontrado=1;
177         }
178         i = i+1;
179     }
180 }
181 if(encontrado==0) {
182     this.modeloListaNoTerminales.add(this.
183         modeloListaNoTerminales.getSize(),elemento);
184     this.jList1.setModel(this.modeloListaNoTerminales);
185     this.jTextField1.requestFocus();
186     this.jTextField1.setText("");
187 }//GEN-LAST:event_jButtonInsertarActionPerformed
188
189 private void jButtonEliminarActionPerformed(java.awt.event.
190     ActionEvent evt) {//GEN-FIRST:
191     event_jButtonEliminarActionPerformed
192     DefaultListModel modelo= null;
193     DefaultListModel producciones = this.panelPadre.getProducciones
194     ();
195     String[] pr;
196     ArrayList<String> prod = new ArrayList();
197     String nt;
198     int seleccion= 0;
199     nt = this.jList1.getSelectedValue().toString();
200
201     seleccion=this.jList1.getSelectedIndex();
202     if(seleccion != -1) {
203         if(producciones != null){
204             int i=0;
205             while(i< producciones.size()){
206                 int j=0;
207                 pr = (producciones.getElementAt(i).toString()).
208                     split(" ");
209                 while(j < pr.length){
210                     if(pr[j].equals(this.jList1.getSelectedValue())){
211                         prod.add(producciones.getElementAt(i).
212                             toString());
213                     }
214                     j++;
215                 }
216                 i++;
217             }
218         }
219         if(prod.size()>0){
220             int conf = JOptionPane.showConfirmDialog(null, "El
221                 símbolo "+nt+" ha sido usado, si lo elimina
222                 también se eliminarán las producciones en las que

```

```

        aparezca, "¿desea eliminarlo?", "Símbolo usado",
        JOptionPane.YES.NO.OPTION);
217     if (conf==0){
218         int k=0;
219         while(k < prod.size()){
220             int l=0;
221             while(l < producciones.size()){
222                 if(prod.get(k).equals(producciones.get(l).
                    toString())){
223                     producciones.remove(l);
224                     break;
225                 }else
226                     l++;
227             }
228             k++;
229         }
230
231         this.panelPadre.setProducciones(producciones);
232         modelo=(DefaultListModel) this.jList1.getModel();
233         modelo.remove(seleccion);
234     }
235
236     }else{
237         modelo=(DefaultListModel) this.jList1.getModel();
238         modelo.remove(seleccion);
239     }
240 }
241 }//GEN-LAST:event_jButtonEliminarActionPerformed
242
243 private void jButtonAceptarActionPerformed(java.awt.event.
    ActionEvent evt) { //GEN-FIRST:
    event_jButtonAceptarActionPerformed
244     if (this.panelPadre!= null) {
245         if (this.jList1.getModel().getSize() > 0) {
246             this.panelPadre.asignarListaSimbolosNoTerminales((
                DefaultListModel) this.jList1.getModel());
247         }
248     }else{
249         this.gramatica.setNoTerminales(modeloListaNoTerminales);
250     }
251     this.dispose();
252 } //GEN-LAST:event_jButtonAceptarActionPerformed
253
254 /**
255  * @param args the command line arguments
256  */
257 public static void main(String args[]) {
258     /* Set the Nimbus look and feel */
259     //<editor-fold defaultstate="collapsed" desc="Look and feel
        setting code (optional) ">
260     /* If Nimbus (introduced in Java SE 6) is not available, stay
        with the default look and feel.

```



```

261      * For details see http://download.oracle.com/javase/tutorial/
      * uiswing/lookandfeel/plaf.html
262      */
263      try {
264          for (javax.swing.UIManager.LookAndFeelInfo info : javax.
              swing.UIManager.getInstalledLookAndFeels()) {
265              if ("Nimbus".equals(info.getName())) {
266                  javax.swing.UIManager.setLookAndFeel(info.
                      getClassName());
267                  break;
268              }
269          }
270      } catch (ClassNotFoundException ex) {
271          java.util.logging.Logger.getLogger(
              VentanaSimbolosNoTerminales.class.getName()).log(java.
              util.logging.Level.SEVERE, null, ex);
272      } catch (InstantiationException ex) {
273          java.util.logging.Logger.getLogger(
              VentanaSimbolosNoTerminales.class.getName()).log(java.
              util.logging.Level.SEVERE, null, ex);
274      } catch (IllegalAccessException ex) {
275          java.util.logging.Logger.getLogger(
              VentanaSimbolosNoTerminales.class.getName()).log(java.
              util.logging.Level.SEVERE, null, ex);
276      } catch (javax.swing.UnsupportedLookAndFeelException ex) {
277          java.util.logging.Logger.getLogger(
              VentanaSimbolosNoTerminales.class.getName()).log(java.
              util.logging.Level.SEVERE, null, ex);
278      }
279      //</editor-fold>
280
281      /* Create and display the form */
282      java.awt.EventQueue.invokeLater(new Runnable() {
283          public void run() {
284              // new VentanaSimbolosNoTerminales().setVisible(true);
285          }
286      });
287  }
288  // Variables declaration - do not modify//GEN-BEGIN:variables
289  private javax.swing.JButton jButtonAceptar;
290  private javax.swing.JButton jButtonCancelar;
291  private javax.swing.JButton jButtonEliminar;
292  private javax.swing.JButton jButtonInsertar;
293  private javax.swing.JList jList1;
294  private javax.swing.JPanel jPanel1;
295  private javax.swing.JScrollPane jScrollPane2;
296  private javax.swing.JTextField jTextField1;
297  // End of variables declaration//GEN-END:variables
298 }

```

### 2.4.24. VentanaSimbolosTerminales.java

```

1 // SimAS / Editor
2 // Ventana Simbolos Terminales
3
4 package es.uco.simas.editor;
5
6 import java.util.ArrayList;
7 import javax.swing.JOptionPane;
8 import javax.swing.DefaultListModel;
9 import javax.swing.ListModel;
10
11 /**
12  * @author vanesa
13  */
14 public class VentanaSimbolosTerminales extends javax.swing.JFrame {
15
16     private DefaultListModel modeloListaTerminales ;
17     private DefaultListModel modeloListaNoTerminales ;
18     private PanelCreacionGramaticaPaso2 panelPadre ;
19
20
21     public VentanaSimbolosTerminales(ListModel simbolosTerminales ,
22         ListModel simbolosNoTerminales , PanelCreacionGramaticaPaso2
23         ventanaPadre) {
24
25         int i= 0;
26
27         DefaultListModel list = new DefaultListModel();
28         this.modeloListaTerminales =list;
29         i=0;
30         while(true){
31             if(i >= simbolosTerminales.getSize()) {
32                 break;
33             }
34             if(i < simbolosTerminales.getSize()) {
35                 this.modeloListaTerminales.add(i,simbolosTerminales.
36                     getElementAt(i));
37                 i = i+1;
38             }
39         }
40         DefaultListModel lista = new DefaultListModel();
41         this.modeloListaNoTerminales =lista;
42         i=0;
43         while(true) {
44             if(i >= simbolosNoTerminales.getSize()) {
45                 break;
46             }
47             if(i < simbolosNoTerminales.getSize()) {
48                 this.modeloListaNoTerminales.add(i,simbolosNoTerminales.
49                     getElementAt(i));
50                 i = i+1;
51             }
52         }
53     }
54 }

```

```

48     }
49
50     initComponents();
51     this.jList1.setModel(simbolosTerminales);
52
53     this.setResizable(false);
54     this.panelPadre = ventanaPadre;
55     this.jTextField1.setText("");
56     // this.btnTerminal13.setText("\u03b5");
57     // this.btnTerminal13.setToolTipText("<html>Inserta el s\u00edmbolo
    <b>\u03b5</b>.</html>");
58     // this.txtSimboloTerminal.requestFocus();
59     // this.estaVentana = this;
60     /* VentanaSimbolosTerminales prueba = new VentanaSimbolosTerminales(
        this, null, null);
61     this.jTextField1.addKeyListener(prueba);
62     VentanaGestionarSimbolosTerminales$4 JdecGenerated190 = new
        VentanaGestionarSimbolosTerminales$4(this);
63     this.txtSimboloTerminal.addActionListener(JdecGenerated190);
64     return;*/
65 }
66 /**
67  * This method is called from within the constructor to initialize
        the form.
68  * WARNING: Do NOT modify this code. The content of this method is
        always
69  * regenerated by the Form Editor.
70  */
71 // @SuppressWarnings("unchecked")
72 // <editor-fold defaultstate="collapsed" desc="Generated Code"> //
    GEN-BEGIN: initComponents
73 private void initComponents() {
74
75     jPanel1 = new javax.swing.JPanel();
76     jButtonAceptar = new javax.swing.JButton();
77     jButtonCancelar = new javax.swing.JButton();
78     jLabel1 = new javax.swing.JLabel();
79     jButton1 = new javax.swing.JButton();
80     jButton2 = new javax.swing.JButton();
81     jButton3 = new javax.swing.JButton();
82     jButton4 = new javax.swing.JButton();
83     jButton5 = new javax.swing.JButton();
84     jButton6 = new javax.swing.JButton();
85     jButton7 = new javax.swing.JButton();
86     jButton8 = new javax.swing.JButton();
87     jButton9 = new javax.swing.JButton();
88     jButton10 = new javax.swing.JButton();
89     jButton11 = new javax.swing.JButton();
90     jButton12 = new javax.swing.JButton();
91     jButton13 = new javax.swing.JButton();
92     jTextField1 = new javax.swing.JTextField();
93     jButtonInsertar = new javax.swing.JButton();
94     jScrollPane2 = new javax.swing.JScrollPane();

```

```

95      jList1 = new javax.swing.JList();
96      jButtonEliminar = new javax.swing.JButton();
97
98      setDefaultCloseOperation(javax.swing.WindowConstants.
        DISPOSE_ON_CLOSE);
99      setTitle("S mbolos Terminales");
100
101      jPanel1.setBackground(new java.awt.Color(233, 244, 244));
102
103      jButtonAceptar.setText("Aceptar");
104      jButtonAceptar.addActionListener(new java.awt.event.
        ActionListener() {
105          public void actionPerformed(java.awt.event.ActionEvent evt)
            {
106              jButtonAceptarActionPerformed(evt);
107          }
108      });
109
110      jButtonCancelar.setText("Cancelar");
111      jButtonCancelar.addActionListener(new java.awt.event.
        ActionListener() {
112          public void actionPerformed(java.awt.event.ActionEvent evt)
            {
113              jButtonCancelarActionPerformed(evt);
114          }
115      });
116
117      jLabel1.setText("S mbolos Terminales Predefinidos");
118
119      jButton1.setText(" + ");
120      jButton1.addActionListener(new java.awt.event.ActionListener()
        {
121          public void actionPerformed(java.awt.event.ActionEvent evt)
            {
122              jButton1ActionPerformed(evt);
123          }
124      });
125
126      jButton2.setText(" - ");
127      jButton2.addActionListener(new java.awt.event.ActionListener()
        {
128          public void actionPerformed(java.awt.event.ActionEvent evt)
            {
129              jButton2ActionPerformed(evt);
130          }
131      });
132
133      jButton3.setText(" * ");
134      jButton3.addActionListener(new java.awt.event.ActionListener()
        {
135          public void actionPerformed(java.awt.event.ActionEvent evt)
            {
136              jButton3ActionPerformed(evt);

```

```
137     }
138   });
139
140   jButton4.setText(" / ");
141   jButton4.addActionListener(new java.awt.event.ActionListener()
142   {
143       public void actionPerformed(java.awt.event.ActionEvent evt)
144       {
145           jButton4ActionPerformed(evt);
146       }
147   });
148
149   jButton5.setText(" ( ");
150   jButton5.addActionListener(new java.awt.event.ActionListener()
151   {
152       public void actionPerformed(java.awt.event.ActionEvent evt)
153       {
154           jButton5ActionPerformed(evt);
155       }
156   });
157
158   jButton6.setText(" ) ");
159   jButton6.addActionListener(new java.awt.event.ActionListener()
160   {
161       public void actionPerformed(java.awt.event.ActionEvent evt)
162       {
163           jButton6ActionPerformed(evt);
164       }
165   });
166
167   jButton7.setText(" { ");
168   jButton7.addActionListener(new java.awt.event.ActionListener()
169   {
170       public void actionPerformed(java.awt.event.ActionEvent evt)
171       {
172           jButton7ActionPerformed(evt);
173       }
174   });
175
176   jButton8.setText(" } ");
177   jButton8.addActionListener(new java.awt.event.ActionListener()
178   {
179       public void actionPerformed(java.awt.event.ActionEvent evt)
180       {
```

```

178         jButton9ActionPerformed(evt);
179     }
180 });
181
182 jButton10.setText(" . ");
183 jButton10.addActionListener(new java.awt.event.ActionListener()
184     {
185         public void actionPerformed(java.awt.event.ActionEvent evt)
186         {
187             jButton10ActionPerformed(evt);
188         }
189     });
190
191 jButton11.setText(" , ");
192 jButton11.addActionListener(new java.awt.event.ActionListener()
193     {
194         public void actionPerformed(java.awt.event.ActionEvent evt)
195         {
196             jButton11ActionPerformed(evt);
197         }
198     });
199
200 jButton12.setText(" ; ");
201 jButton12.addActionListener(new java.awt.event.ActionListener()
202     {
203         public void actionPerformed(java.awt.event.ActionEvent evt)
204         {
205             jButton12ActionPerformed(evt);
206         }
207     });
208
209 jButton13.setText(" : ");
210 jButton13.addActionListener(new java.awt.event.ActionListener() {
211     public void actionPerformed(java.awt.event.ActionEvent evt)
212     {
213         jButton13ActionPerformed(evt);
214     }
215 });
216
217 jButtonInsertar.setText(" Insertar");
218 jButtonInsertar.addActionListener(new java.awt.event.
219     ActionListener() {
220         public void actionPerformed(java.awt.event.ActionEvent evt)
221         {
222             jButtonInsertarActionPerformed(evt);
223         }
224     });
225
226 jList1.setSelectionMode(javax.swing.ListSelectionModel.
227     SINGLE_SELECTION);
228 jList1.setDropMode(javax.swing.DropMode.INSERT);
229 jScrollPane2.setViewportView(jList1);

```

```

220
221     JButtonEliminar.setText("Eliminar");
222     JButtonEliminar.addActionListener(new java.awt.event.
        ActionListener() {
223         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
224             JButtonEliminarActionPerformed(evt);
225         }
226     });
227
228     javax.swing.GroupLayout jPanel1Layout = new javax.swing.
        GroupLayout(jPanel1);
229     jPanel1.setLayout(jPanel1Layout);
230     jPanel1Layout.setHorizontalGroup(
231         jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
            Alignment.LEADING)
232         .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            jPanel1Layout.createSequentialGroup()
233             .addGap(javax.swing.GroupLayout.DEFAULT_SIZE,
                Short.MAX_VALUE)
234             .addComponent(JButtonCancelar)
235             .addGap(18, 18, 18)
236             .addComponent(JButtonAceptar)
237             .addGap(20, 20, 20))
238         .addGroup(jPanel1Layout.createSequentialGroup()
239             .addGap(29, 29, 29)
240             .addGroup(jPanel1Layout.createParallelGroup(javax.swing
                GroupLayout.Alignment.LEADING, false)
                .addComponent(JLabel1)
241                .addGroup(jPanel1Layout.createSequentialGroup()
242                    .addGroup(jPanel1Layout.createParallelGroup(
243                        javax.swing.GroupLayout.Alignment.TRAILING,
244                        false)
245                        .addComponent(JTextField1, javax.swing.
                            GroupLayout.Alignment.LEADING)
246                        .addGroup(jPanel1Layout.
                            createSequentialGroup()
247                                .addComponent(JButton1)
248                                .addPreferredGap(LayoutStyle.ComponentPlacement.
                                    RELATED)
249                                .addComponent(JButton2)
250                                .addPreferredGap(LayoutStyle.ComponentPlacement.
                                    RELATED)
251                                .addComponent(JButton3)
252                                .addPreferredGap(LayoutStyle.ComponentPlacement.
                                    RELATED)
253                                .addComponent(JButton4)
254                                .addPreferredGap(LayoutStyle.ComponentPlacement.
                                    RELATED)

```

```

254 |         .addComponent(jButton5)
255 |         .addPreferredGap(javax.swing.
           |             LayoutStyle.ComponentPlacement.
           |             RELATED)
256 |         .addComponent(jButton6)
257 |         .addPreferredGap(javax.swing.
           |             LayoutStyle.ComponentPlacement.
           |             RELATED)
258 |         .addComponent(jButton7)))
259 |     .addGroup(jPanel1Layout.createParallelGroup(
           |         javax.swing.GroupLayout.Alignment.LEADING)
260 |         .addGroup(jPanel1Layout.
           |             createSequentialGroup())
261 |         .addPreferredGap(javax.swing.
           |             LayoutStyle.ComponentPlacement.
           |             RELATED)
262 |         .addComponent(jButton8)
263 |         .addPreferredGap(javax.swing.
           |             LayoutStyle.ComponentPlacement.
           |             RELATED)
264 |         .addComponent(jButton9)
265 |         .addPreferredGap(javax.swing.
           |             LayoutStyle.ComponentPlacement.
           |             RELATED)
266 |         .addComponent(jButton10)
267 |         .addPreferredGap(javax.swing.
           |             LayoutStyle.ComponentPlacement.
           |             RELATED)
268 |         .addComponent(jButton11)
269 |         .addPreferredGap(javax.swing.
           |             LayoutStyle.ComponentPlacement.
           |             RELATED)
270 |         .addComponent(jButton12)
271 |         .addPreferredGap(javax.swing.
           |             LayoutStyle.ComponentPlacement.
           |             RELATED)
272 |         .addComponent(jButton13))
273 |     .addGroup(jPanel1Layout.
           |         createSequentialGroup())
274 |         .addGap(23, 23, 23)
275 |         .addComponent(jButtonInsertar)
276 |         .addPreferredGap(javax.swing.
           |             LayoutStyle.ComponentPlacement.
           |             RELATED)
277 |         .addComponent(jButtonEliminar)))
278 |     .addComponent(jScrollPane2))
279 |     .addContainerGap(41, Short.MAX_VALUE))
280 | );
281 | jPanel1Layout.setVerticalGroup(
282 |     jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
           |         Alignment.LEADING)
283 |     .addGroup(jPanel1Layout.createSequentialGroup())
284 |     .addContainerGap()

```



```

285         .addComponent(jLabel1)
286         .addPreferredGap(javax.swing.LayoutStyle.
           ComponentPlacement.UNRELATED)
287         .addGroup(jPanel1Layout.createParallelGroup(javax.swing
           .GroupLayout.Alignment.BASELINE)
288             .addComponent(jButton1)
289             .addComponent(jButton2)
290             .addComponent(jButton3)
291             .addComponent(jButton4)
292             .addComponent(jButton5)
293             .addComponent(jButton6)
294             .addComponent(jButton7)
295             .addComponent(jButton8)
296             .addComponent(jButton9)
297             .addComponent(jButton10)
298             .addComponent(jButton11)
299             .addComponent(jButton12)
300             .addComponent(jButton13))
301         .addGap(36, 36, 36)
302         .addGroup(jPanel1Layout.createParallelGroup(javax.swing
           .GroupLayout.Alignment.BASELINE)
303             .addComponent(jTextField1, javax.swing.GroupLayout.
               PREFERRED_SIZE, javax.swing.GroupLayout.
               DEFAULT_SIZE, javax.swing.GroupLayout.
               PREFERRED_SIZE)
304             .addComponent(jButtonInsertar)
305             .addComponent(jButtonEliminar))
306         .addPreferredGap(javax.swing.LayoutStyle.
           ComponentPlacement.RELATED)
307         .addComponent(jScrollPane2, javax.swing.GroupLayout.
           PREFERRED_SIZE, 216, javax.swing.GroupLayout.
           PREFERRED_SIZE)
308         .addPreferredGap(javax.swing.LayoutStyle.
           ComponentPlacement.RELATED, 22, Short.MAX_VALUE)
309         .addGroup(jPanel1Layout.createParallelGroup(javax.swing
           .GroupLayout.Alignment.BASELINE)
310             .addComponent(jButtonAceptar)
311             .addComponent(jButtonCancelar))
312         .addGap(21, 21, 21))
313     );
314
315     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
           getContentPane());
316     getContentPane().setLayout(layout);
317     layout.setHorizontalGroup(
318         layout.createParallelGroup(javax.swing.GroupLayout.
           Alignment.LEADING)
319         .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE
           , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE
           )
320     );
321     layout.setVerticalGroup(

```

```

322         layout.createParallelGroup(javax.swing.GroupLayout.
323             Alignment.LEADING)
324             .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE
325                 , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE
326                 );
327     pack();
328 }// </editor-fold> //GEN-END: initComponents
329 private void jButtonCancelarActionPerformed(java.awt.event.
330     ActionEvent evt) { //GEN-FIRST:
331         event_jButtonCancelarActionPerformed
332         int conf = JOptionPane.showConfirmDialog(null, "¿Desea salir
333             de la edicion de la gramática?", "Salir",JOptionPane.
334             YES.NO_OPTION);
335
336         if (conf==0)
337             this.dispose();
338 }//GEN-LAST: event_jButtonCancelarActionPerformed
339
340 private void jButtonInsertarActionPerformed(java.awt.event.
341     ActionEvent evt) { //GEN-FIRST:
342         event_jButtonInsertarActionPerformed
343         /* DefaultListModel listModel = new DefaultListModel();
344         String elemento = jTextField1.getText();
345         listModel.addElement(elemento);*/
346         String elemento = jTextField1.getText();
347         int encontrado= 0;
348         int i= 0;
349
350         if (elemento.equals("")){
351             encontrado =1;
352         }else{
353             while(true) {
354                 if(i >= this.modeloListaTerminales.getSize()) {
355                     break;
356                 }
357                 if(i < this.modeloListaTerminales.getSize()) {
358                     if(this.modeloListaTerminales.elementAt(i).equals(
359                         elemento)) {
360                         encontrado=1;
361                     }
362                     i = i+1;
363                     //continue ;
364                 }
365             }
366         }
367         if (encontrado==0)
368         {

```

```

364         this.modeloListaTerminales.add(this.modeloListaTerminales.getSize
365         (), elemento);
366         this.jList1.setModel(this.modeloListaTerminales);
367     }
368     this.jTextField1.requestFocus();
369     this.jTextField1.setText("");
370 } //GEN-LAST: event_jButtonInsertarActionPerformed
371
372 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt
373 ) { //GEN-FIRST: event_jButton1ActionPerformed
374     int encontrado= 0;
375     int i= 0;
376     encontrado=0;
377     i=0;
378     while(true) {
379         if(i >= this.modeloListaTerminales.getSize()) {
380             break;
381         }
382         if(i < this.modeloListaTerminales.getSize()) {
383             if(this.modeloListaTerminales.elementAt(i).equals("+
384             ")) {
385                 encontrado=1;
386             }
387             i = i+1;
388         }
389     }
390     if(encontrado==0) {
391         this.modeloListaTerminales.add(this.modeloListaTerminales.
392         getSize(), "+");
393         this.jList1.setModel(this.modeloListaTerminales);
394     }
395     this.jTextField1.requestFocus();
396 } //GEN-LAST: event_jButton1ActionPerformed
397
398 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt
399 ) { //GEN-FIRST: event_jButton2ActionPerformed
400     int encontrado= 0;
401     int i= 0;
402     while(true) {
403         if(i >= this.modeloListaTerminales.getSize()) {
404             break;
405         }
406         if(i < this.modeloListaTerminales.getSize()) {
407             if(this.modeloListaTerminales.elementAt(i).equals("-
408             ")) {
409                 encontrado=1;
410             }
411             i = i+1;
412         }
413     }
414     if(encontrado==0) {

```

```

410         this.modeloListaTerminales.add(this.modeloListaTerminales.
411             getSize(),"-");
412         this.jList1.setModel(this.modeloListaTerminales);
413     }
414     this.jTextField1.requestFocus();
415 }//GEN-LAST:event_jButton2ActionPerformed
416
417 private void jButtonAceptarActionPerformed(java.awt.event.
418     ActionEvent evt) {//GEN-FIRST:
419     event_jButtonAceptarActionPerformed
420     if(this.panelPadre!= null) {
421         if(this.jList1.getModel().getSize() > 0) {
422             this.panelPadre.asignarListaSimbolosTerminales((
423                 DefaultListModel)this.jList1.getModel());
424         }
425     }
426     this.dispose();
427 }//GEN-LAST:event_jButtonAceptarActionPerformed
428
429 private void jButtonEliminarActionPerformed(java.awt.event.
430     ActionEvent evt) {//GEN-FIRST:
431     event_jButtonEliminarActionPerformed
432     DefaultListModel modelo= null;
433     DefaultListModel producciones = this.panelPadre.getProducciones
434     ();
435     String[] pr;
436     ArrayList<String> prod = new ArrayList();
437     String nt;
438     int seleccion= 0;
439     nt = this.jList1.getSelectedValue().toString();
440
441     seleccion=this.jList1.getSelectedIndex();
442     if(seleccion != -1) {
443         if(producciones != null){
444             int i=0;
445             while(i< producciones.size()){
446                 int j=0;
447                 pr = (producciones.elementAt(i).toString()).
448                     split(" ");
449                 while(j < pr.length){
450                     if(pr[j].equals(this.jList1.getSelectedValue())
451                     ){
452                         prod.add(producciones.elementAt(i).
453                             toString());
454                     }
455                     j++;
456                 }
457                 i++;
458             }
459         }
460     }
461     if(prod.size()>0){

```

```

452         int conf = JOptionPane.showConfirmDialog(null, "El
        símbolo "+nt+" ha sido usado, si lo elimina
        también se eliminarán las producciones en las que
        aparezca, ¿desea eliminarlo?", "Símbolo usado",
        JOptionPane.YES_NO_OPTION);
453     if (conf==0){
454         int k=0;
455         while(k < prod.size()){
456             int l=0;
457             while(l < producciones.size()){
458                 if(prod.get(k).equals(producciones.get(l).
                    toString())){
459                     producciones.remove(l);
460                     break;
461                 } else
462                     l++;
463             }
464             k++;
465         }
466
467         this.panelPadre.setProducciones(producciones);
468         modelo=(DefaultListModel) this.jList1.getModel();
469         modelo.remove(seleccion);
470     }
471
472     } else {
473         modelo=(DefaultListModel) this.jList1.getModel();
474         modelo.remove(seleccion);
475     }
476 }
477
478
479
480 /*DefaultListModel modelo= null;
481 int seleccion= 0;
482 seleccion=this.jList1.getSelectedIndex();
483
484 if(seleccion != -1) {
485     modelo=(DefaultListModel) this.jList1.getModel();
486     modelo.remove(seleccion);
487 */
488 //GEN-LAST: event_jButtonEliminarActionPerformed
489
490 private void jButton3ActionPerformed(java.awt.event.ActionEvent evt
    ) //GEN-FIRST: event_jButton3ActionPerformed
491     int encontrado= 0;
492     int i= 0;
493
494     while(true) {
495         if(i >= this.modeloListaTerminales.getSize()) {
496             break;
497         }
498         if(i < this.modeloListaTerminales.getSize()) {

```

```

499         if (this.modeloListaTerminales.getElementAt(i).equals("*
500             ")) {
501             encontrado=1;
502         }
503         i = i+1;
504     }
505     if (encontrado==0){
506         this.modeloListaTerminales.add(this.modeloListaTerminales.
507             getSize(),"*");
508         this.jList1.setModel(this.modeloListaTerminales);
509     }
510     this.jTextField1.requestFocus();
511 }//GEN-LAST:event_jButton3ActionPerformed
512
513 private void jButton4ActionPerformed(java.awt.event.ActionEvent evt
514 ) { //GEN-FIRST:event_jButton4ActionPerformed
515     int encontrado= 0;
516     int i= 0;
517
518     while(true) {
519         if(i >= this.modeloListaTerminales.getSize()){
520             break;
521         }
522         if(i < this.modeloListaTerminales.getSize()) {
523             if (this.modeloListaTerminales.getElementAt(i).equals("/
524                 ")) {
525                 encontrado=1;
526             }
527             i = i+1;
528         }
529     }
530     if (encontrado==0) {
531         this.modeloListaTerminales.add(this.modeloListaTerminales.
532             getSize(),"/");
533         this.jList1.setModel(this.modeloListaTerminales);
534     }
535     this.jTextField1.requestFocus();
536 }//GEN-LAST:event_jButton4ActionPerformed
537
538 private void jButton5ActionPerformed(java.awt.event.ActionEvent evt
539 ) { //GEN-FIRST:event_jButton5ActionPerformed
540     int encontrado= 0;
541     int i= 0;
542
543     while(true) {
544         if(i >= this.modeloListaTerminales.getSize()){
545             break;
546         }
547         if(i < this.modeloListaTerminales.getSize()) {
548             if (this.modeloListaTerminales.getElementAt(i).equals("("
549                 ")) {
550                 encontrado=1;

```

```

545         }
546         i = i+1;
547     }
548 }
549 if(encontrado==0) {
550     this.modeloListaTerminales.add(this.modeloListaTerminales.
551         getSize(),"");
552     this.jList1.setModel(this.modeloListaTerminales);
553 }
554 this.jTextField1.requestFocus();
555 }//GEN-LAST:event_jButton5ActionPerformed
556
557 private void jButton6ActionPerformed(java.awt.event.ActionEvent evt
558 ) { //GEN-FIRST:event_jButton6ActionPerformed
559     int encontrado= 0;
560     int i= 0;
561
562     while(true) {
563         if(i >= this.modeloListaTerminales.getSize()) {
564             break;
565         }
566         if(i < this.modeloListaTerminales.getSize()) {
567             if(this.modeloListaTerminales.elementAt(i).equals("
568                 ")) {
569                 encontrado=1;
570             }
571             i = i+1;
572         }
573     }
574     if(encontrado==0) {
575         this.modeloListaTerminales.add(this.modeloListaTerminales.
576             getSize(),"");
577         this.jList1.setModel(this.modeloListaTerminales);
578     }
579     this.jTextField1.requestFocus();
580 }//GEN-LAST:event_jButton6ActionPerformed
581
582 private void jButton7ActionPerformed(java.awt.event.ActionEvent evt
583 ) { //GEN-FIRST:event_jButton7ActionPerformed
584     int encontrado= 0;
585     int i= 0;
586
587     while(true) {
588         if(i >= this.modeloListaTerminales.getSize()) {
589             break;
590         }
591         if(i < this.modeloListaTerminales.getSize()) {
592             if(this.modeloListaTerminales.elementAt(i).equals("
593                 ")) {
594                 encontrado=1;
595             }
596             i = i+1;
597         }
598     }

```

```

592     }
593     if(encontrado==0){
594         this.modeloListaTerminales.add(this.modeloListaTerminales.
595             getSize(),"");
596         this.jList1.setModel(this.modeloListaTerminales);
597     }
598     this.jTextField1.requestFocus();
599 }//GEN-LAST:event_jButton7ActionPerformed
600
601 private void jButton8ActionPerformed(java.awt.event.ActionEvent evt
602 ) { //GEN-FIRST:event_jButton8ActionPerformed
603     int encontrado= 0;
604     int i= 0;
605
606     while(true) {
607         if(i >= this.modeloListaTerminales.getSize()) {
608             break;
609         }
610         if(i < this.modeloListaTerminales.getSize()) {
611             if(this.modeloListaTerminales.elementAt(i).equals("}
612                 ")) {
613                 encontrado=1;
614             }
615             i = i+1;
616         }
617     }
618     if(encontrado==0) {
619         this.modeloListaTerminales.add(this.modeloListaTerminales.
620             getSize(),"");
621         this.jList1.setModel(this.modeloListaTerminales);
622     }
623     this.jTextField1.requestFocus();
624 }//GEN-LAST:event_jButton8ActionPerformed
625
626 private void jButton9ActionPerformed(java.awt.event.ActionEvent evt
627 ) { //GEN-FIRST:event_jButton9ActionPerformed
628     int encontrado= 0;
629     int i= 0;
630
631     while(true) {
632         if(i >= this.modeloListaTerminales.getSize()) {
633             break;
634         }
635         if(i < this.modeloListaTerminales.getSize()) {
636             if(this.modeloListaTerminales.elementAt(i).equals("="
637                 ")) {

```



```

638         this.modeloListaTerminales.add(this.modeloListaTerminales.
639             getSize(), "=");
640     }
641     this.jTextField1.requestFocus();
642 } //GEN-LAST:event_jButton9ActionPerformed
643
644 private void jButton10ActionPerformed(java.awt.event.ActionEvent
645     evt) { //GEN-FIRST:event_jButton10ActionPerformed
646     int encontrado= 0;
647     int i= 0;
648
649     while(true) {
650         if(i >= this.modeloListaTerminales.getSize()) {
651             break;
652         }
653         if(i < this.modeloListaTerminales.getSize()) {
654             if(this.modeloListaTerminales.elementAt(i).equals("
655                 ")) {
656                 encontrado=1;
657             }
658             i = i+1;
659         }
660     }
661     if(encontrado==0) {
662         this.modeloListaTerminales.add(this.modeloListaTerminales.
663             getSize(), ".");
664         this.jList1.setModel(this.modeloListaTerminales);
665     }
666     this.jTextField1.requestFocus();
667 } //GEN-LAST:event_jButton10ActionPerformed
668
669 private void jButton11ActionPerformed(java.awt.event.ActionEvent
670     evt) { //GEN-FIRST:event_jButton11ActionPerformed
671     int encontrado= 0;
672     int i= 0;
673
674     while(true) {
675         if(i >= this.modeloListaTerminales.getSize()) {
676             break;
677         }
678         if(i < this.modeloListaTerminales.getSize()) {
679             if(this.modeloListaTerminales.elementAt(i).equals(",
680                 ")) {
681                 encontrado=1;
682             }
683             i = i+1;
684         }
685     }
686     if(encontrado==0) {
687         this.modeloListaTerminales.add(this.modeloListaTerminales.
688             getSize(), ",");
689         this.jList1.setModel(this.modeloListaTerminales);

```

```

684     }
685     this.jTextField1.requestFocus();
686 }//GEN-LAST:event_jButton11ActionPerformed
687
688 private void jButton12ActionPerformed(java.awt.event.ActionEvent
689     evt) {//GEN-FIRST:event_jButton12ActionPerformed
690     int encontrado= 0;
691     int i= 0;
692
693     while(true) {
694         if(i >= this.modeloListaTerminales.getSize()) {
695             break;
696         }
697         if(i < this.modeloListaTerminales.getSize()) {
698             if(this.modeloListaTerminales.elementAt(i).equals("
699                 ")) {
700                 encontrado=1;
701             }
702             i = i+1;
703         }
704     }
705     if(encontrado==0) {
706         this.modeloListaTerminales.add(this.modeloListaTerminales.
707             getSize(),":");
708         this.jList1.setModel(this.modeloListaTerminales);
709     }
710     this.jTextField1.requestFocus();
711 }//GEN-LAST:event_jButton12ActionPerformed
712
713 private void jButton13ActionPerformed(java.awt.event.ActionEvent
714     evt) {//GEN-FIRST:event_jButton13ActionPerformed
715     int encontrado= 0;
716     int i= 0;
717
718     while(true) {
719         if(i >= this.modeloListaTerminales.getSize()) {
720             break;
721         }
722         if(i < this.modeloListaTerminales.getSize()) {
723             if(this.modeloListaTerminales.elementAt(i).equals("
724                 ")) {
725                 encontrado=1;
726             }
727             i = i+1;
728         }
729     }
730     if(encontrado==0) {
731         this.modeloListaTerminales.add(this.modeloListaTerminales.
732             getSize(),":");
733         this.jList1.setModel(this.modeloListaTerminales);
734     }
735     this.jTextField1.requestFocus();
736 }//GEN-LAST:event_jButton13ActionPerformed

```

```

731
732  /**
733   * @param args the command line arguments
734   */
735  public static void main(String args[]) {
736      /* Set the Nimbus look and feel */
737      //<editor-fold defaultstate="collapsed" desc=" Look and feel
          setting code (optional) ">
738      /* If Nimbus (introduced in Java SE 6) is not available, stay
          with the default look and feel.
739       * For details see http://download.oracle.com/javase/tutorial/
          uiswing/lookandfeel/plaf.html
740       */
741      try {
742          for (javax.swing.UIManager.LookAndFeelInfo info : javax.
              swing.UIManager.getInstalledLookAndFeels()) {
743              if ("Nimbus".equals(info.getName())) {
744                  javax.swing.UIManager.setLookAndFeel(info.
                      getClassName());
745                  break;
746              }
747          }
748      } catch (ClassNotFoundException ex) {
749          java.util.logging.Logger.getLogger(
              VentanaSimbolosTerminales.class.getName()).log(java.
              util.logging.Level.SEVERE, null, ex);
750      } catch (InstantiationException ex) {
751          java.util.logging.Logger.getLogger(
              VentanaSimbolosTerminales.class.getName()).log(java.
              util.logging.Level.SEVERE, null, ex);
752      } catch (IllegalAccessException ex) {
753          java.util.logging.Logger.getLogger(
              VentanaSimbolosTerminales.class.getName()).log(java.
              util.logging.Level.SEVERE, null, ex);
754      } catch (javax.swing.UnsupportedLookAndFeelException ex) {
755          java.util.logging.Logger.getLogger(
              VentanaSimbolosTerminales.class.getName()).log(java.
              util.logging.Level.SEVERE, null, ex);
756      }
757      //</editor-fold>
758
759      /* Create and display the form */
760      java.awt.EventQueue.invokeLater(new Runnable() {
761          public void run() {
762              //      new VentanaSimbolosTerminales().setVisible(true);
763              }
764          });
765      }
766      // Variables declaration - do not modify//GEN-BEGIN:variables
767      private javax.swing.JButton jButton1;
768      private javax.swing.JButton jButton10;
769      private javax.swing.JButton jButton11;
770      private javax.swing.JButton jButton12;

```

```

771 private javax.swing.JButton jButton13;
772 private javax.swing.JButton jButton2;
773 private javax.swing.JButton jButton3;
774 private javax.swing.JButton jButton4;
775 private javax.swing.JButton jButton5;
776 private javax.swing.JButton jButton6;
777 private javax.swing.JButton jButton7;
778 private javax.swing.JButton jButton8;
779 private javax.swing.JButton jButton9;
780 private javax.swing.JButton jButtonAceptar;
781 private javax.swing.JButton jButtonCancelar;
782 private javax.swing.JButton jButtonEliminar;
783 private javax.swing.JButton jButtonInsertar;
784 private javax.swing.JLabel jLabel1;
785 private javax.swing.JList jList1;
786 private javax.swing.JPanel jPanel1;
787 private javax.swing.JScrollPane jScrollPane2;
788 private javax.swing.JTextField jTextField1;
789 // End of variables declaration//GEN-END:variables
790 }

```

### 2.4.25. Visualizacion.java

```

1 //SimAS / Editor
2 // Visualizacion
3
4 package es.uco.simas.editor;
5 import es.uco.simas.util.gramatica.*;
6 import javax.swing.*;
7
8 /**
9  * @author vanesa
10  */
11 public class Visualizacion extends javax.swing.JFrame {
12     Gramatica gramatica = new Gramatica();
13     DefaultListModel noTerminales = new DefaultListModel();
14     DefaultListModel terminales = new DefaultListModel();
15
16     public Visualizacion(Gramatica gr) {
17
18         initComponents();
19         this.gramatica = gr;
20
21         this.jListNT.setModel(this.gramatica.getNoTerminales());
22         this.jListT.setModel(this.gramatica.getTerminales());
23         this.jListProd.setModel(this.gramatica.getProducciones());
24         this.nombre.setText(this.gramatica.getNombre());
25         this.descripcion.setText(this.gramatica.getDescripcion());
26         this.inicial.setText(this.gramatica.getSimbInicial());
27     }

```

```

28
29  /**
30   * This method is called from within the constructor to initialize
      the form.
31   * WARNING: Do NOT modify this code. The content of this method is
      always
32   * regenerated by the Form Editor.
33   */
34  @SuppressWarnings("unchecked")
35  // <editor-fold defaultstate="collapsed" desc="Generated Code">//
      GEN-BEGIN: initComponents
36  private void initComponents() {
37
38      jPanel1 = new javax.swing.JPanel();
39      jLabel1 = new javax.swing.JLabel();
40      jLabel3 = new javax.swing.JLabel();
41      jScrollPane1 = new javax.swing.JScrollPane();
42      jListNT = new javax.swing.JList();
43      jLabel5 = new javax.swing.JLabel();
44      jScrollPane2 = new javax.swing.JScrollPane();
45      jListT = new javax.swing.JList();
46      jLabel6 = new javax.swing.JLabel();
47      jLabel8 = new javax.swing.JLabel();
48      jScrollPane3 = new javax.swing.JScrollPane();
49      jListProd = new javax.swing.JList();
50      jLabel9 = new javax.swing.JLabel();
51      nombre = new javax.swing.JLabel();
52      descripcion = new javax.swing.JLabel();
53      jButtonNT = new javax.swing.JButton();
54      inicial = new javax.swing.JLabel();
55
56      setDefaultCloseOperation(javax.swing.WindowConstants.
          DISPOSE_ON_CLOSE);
57
58      jPanel1.setBackground(new java.awt.Color(233, 244, 244));
59
60      jLabel1.setText("Nombre gramática: ");
61
62      jLabel3.setText("Descripcion: ");
63
64      jScrollPane1.setViewportView(jListNT);
65
66      jLabel5.setText("Símbolos No Terminales:");
67
68      jScrollPane2.setViewportView(jListT);
69
70      jLabel6.setText("Símbolos Terminales:");
71
72      jLabel8.setText("Producciones:");
73
74      jScrollPane3.setViewportView(jListProd);
75
76      jLabel9.setText("Símbolo Inicial de la gramática:");

```

```

77
78     jButtonNT.setText("Editar");
79     jButtonNT.addActionListener(new java.awt.event.ActionListener()
80     {
81         public void actionPerformed(java.awt.event.ActionEvent evt)
82         {
83             jButtonNTActionPerformed(evt);
84         }
85     });
86
87     javax.swing.GroupLayout jPanel1Layout = new javax.swing.
88     GroupLayout(jPanel1);
89     jPanel1.setLayout(jPanel1Layout);
90     jPanel1Layout.setHorizontalGroup(
91     jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
92     Alignment.LEADING)
93     .addGroup(jPanel1Layout.createSequentialGroup()
94     .addGap(31, 31, 31)
95     .addGroup(jPanel1Layout.createParallelGroup(javax.swing
96     GroupLayout.Alignment.LEADING)
97     .addGroup(jPanel1Layout.createSequentialGroup()
98     .addComponent(jLabel8)
99     .addGroup(jPanel1Layout.createSequentialGroup()
100     .addComponent(jLabel1)
101     .addPreferredGap(javax.swing.
102     LayoutStyle.ComponentPlacement.
103     RELATED)
104     .addComponent(nombre))
105     .addGroup(jPanel1Layout.createSequentialGroup()
106     .addComponent(jLabel3)
107     .addPreferredGap(javax.swing.
108     LayoutStyle.ComponentPlacement.
109     RELATED)
110     .addComponent(descripcion))
111     .addComponent(jScrollPane1, javax.swing.
112     GroupLayout.DEFAULT_SIZE, 551, Short.
113     MAX_VALUE)
114     .addComponent(jLabel5)
115     .addComponent(jLabel6)
116     .addComponent(jScrollPane2)
117     .addComponent(jScrollPane3))
118     .addGap(18, 18, 18)
119     .addComponent(jButtonNT))
120     .addGroup(jPanel1Layout.createSequentialGroup()
121     .addComponent(jLabel9)
122     .addPreferredGap(javax.swing.LayoutStyle.
123     ComponentPlacement.UNRELATED)
124     .addComponent(inicial)))

```

```

114         .addContainerGap(107, Short.MAX_VALUE))
115     );
116     jPanel1Layout.setVerticalGroup(
117         jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
            Alignment.LEADING)
118     .addGroup(jPanel1Layout.createSequentialGroup())
119     .addGap(22, 22, 22)
120     .addGroup(jPanel1Layout.createParallelGroup(javax.swing
            GroupLayout.Alignment.BASELINE)
121     .addComponent(jLabel1)
122     .addComponent(nombre))
123     .addPreferredGap(javax.swing.LayoutStyle.
            ComponentPlacement.RELATED)
124     .addGroup(jPanel1Layout.createParallelGroup(javax.swing
            GroupLayout.Alignment.BASELINE)
125     .addComponent(jLabel3)
126     .addComponent(descripcion))
127     .addGap(9, 9, 9)
128     .addComponent(jLabel5, javax.swing.GroupLayout.
            PREFERRED_SIZE, 17, javax.swing.GroupLayout.
            PREFERRED_SIZE)
129     .addGroup(jPanel1Layout.createParallelGroup(javax.swing
            GroupLayout.Alignment.LEADING)
130     .addGroup(jPanel1Layout.createSequentialGroup())
131     .addPreferredGap(javax.swing.LayoutStyle.
            ComponentPlacement.RELATED)
132     .addComponent(jScrollPane1, javax.swing.
            GroupLayout.PREFERRED_SIZE, 113, javax.
            swing.GroupLayout.PREFERRED_SIZE))
133     .addGroup(jPanel1Layout.createSequentialGroup())
134     .addGap(46, 46, 46)
135     .addComponent(jButtonNT)))
136     .addPreferredGap(javax.swing.LayoutStyle.
            ComponentPlacement.RELATED)
137     .addComponent(jLabel6)
138     .addGap(9, 9, 9)
139     .addComponent(jScrollPane2, javax.swing.GroupLayout.
            PREFERRED_SIZE, 106, javax.swing.GroupLayout.
            PREFERRED_SIZE)
140     .addGroup(jPanel1Layout.createParallelGroup(javax.swing
            GroupLayout.Alignment.LEADING)
141     .addGroup(jPanel1Layout.createSequentialGroup())
142     .addPreferredGap(javax.swing.LayoutStyle.
            ComponentPlacement.UNRELATED)
143     .addComponent(jLabel8)
144     .addPreferredGap(javax.swing.LayoutStyle.
            ComponentPlacement.RELATED)
145     .addComponent(jScrollPane3, javax.swing.
            GroupLayout.PREFERRED_SIZE, javax.swing.
            GroupLayout.DEFAULT_SIZE, javax.swing.
            GroupLayout.PREFERRED_SIZE)
146     .addGap(24, 24, 24)
147     .addComponent(jLabel9)

```

```

148         .addContainerGap(18, Short.MAX_VALUE))
149     .addGroup(javax.swing.GroupLayout.Alignment.
        TRAILING, jPanel1Layout.createSequentialGroup())
150     .addPreferredGap(javax.swing.LayoutStyle.
        ComponentPlacement.RELATED, javax.swing.
        GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
151     .addComponent(inicial)
152     .addContainerGap()))
153 );
154
155 javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
        getContentPane());
156 getContentPane().setLayout(layout);
157 layout.setHorizontalGroup(
158     layout.createParallelGroup(javax.swing.GroupLayout.
        Alignment.LEADING)
159     .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE
        , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE
        )
160 );
161 layout.setVerticalGroup(
162     layout.createParallelGroup(javax.swing.GroupLayout.
        Alignment.LEADING)
163     .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE
        , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE
        )
164 );
165
166 pack();
167 }// </editor-fold>//GEN-END: initComponents
168
169 private void jButtonNTActionPerformed(java.awt.event.ActionEvent
    evt) { //GEN-FIRST: event_jButtonNTActionPerformed
170     VentanaSimbolosNoTerminales noTerminal = new
        VentanaSimbolosNoTerminales(gramatica.getNoTerminales(),
        gramatica.getTerminales(), null, gramatica);
171     noTerminal.setVisible(true);
172     noTerminal.setLocationRelativeTo(null);
173
174 } //GEN-LAST: event_jButtonNTActionPerformed
175
176 /**
177  * @param args the command line arguments
178  */
179 public static void main(String args[]) {
180     /* Set the Nimbus look and feel */
181     //<editor-fold defaultstate="collapsed" desc="Look and feel
        setting code (optional) ">
182     /* If Nimbus (introduced in Java SE 6) is not available, stay
        with the default look and feel.
183     * For details see http://download.oracle.com/javase/tutorial/
        uiswing/lookandfeel/plaf.html
184     */

```



```

185     try {
186         for (javax.swing.UIManager.LookAndFeelInfo info : javax.
            swing.UIManager.getInstalledLookAndFeels()) {
187             if ("Nimbus".equals(info.getName())) {
188                 javax.swing.UIManager.setLookAndFeel(info.
                    getClassName());
189                 break;
190             }
191         }
192     } catch (ClassNotFoundException ex) {
193         java.util.logging.Logger.getLogger(Visualizacion.class.
            getName()).log(java.util.logging.Level.SEVERE, null, ex
            );
194     } catch (InstantiationException ex) {
195         java.util.logging.Logger.getLogger(Visualizacion.class.
            getName()).log(java.util.logging.Level.SEVERE, null, ex
            );
196     } catch (IllegalAccessException ex) {
197         java.util.logging.Logger.getLogger(Visualizacion.class.
            getName()).log(java.util.logging.Level.SEVERE, null, ex
            );
198     } catch (javax.swing.UnsupportedLookAndFeelException ex) {
199         java.util.logging.Logger.getLogger(Visualizacion.class.
            getName()).log(java.util.logging.Level.SEVERE, null, ex
            );
200     }
201     //</editor-fold>
202
203     /* Create and display the form */
204     java.awt.EventQueue.invokeLater(new Runnable() {
205         public void run() {
206             // new Visualizacion().setVisible(true);
207         }
208     });
209 }
210 // Variables declaration - do not modify//GEN-BEGIN:variables
211 private javax.swing.JLabel descripcion;
212 private javax.swing.JLabel inicial;
213 private javax.swing.JButton jButtonNT;
214 private javax.swing.JLabel jLabel1;
215 private javax.swing.JLabel jLabel3;
216 private javax.swing.JLabel jLabel5;
217 private javax.swing.JLabel jLabel6;
218 private javax.swing.JLabel jLabel8;
219 private javax.swing.JLabel jLabel9;
220 private javax.swing.JList jListNT;
221 private javax.swing.JList jListProd;
222 private javax.swing.JList jListT;
223 private javax.swing.JPanel jPanel1;
224 private javax.swing.JScrollPane jScrollPane1;
225 private javax.swing.JScrollPane jScrollPane2;
226 private javax.swing.JScrollPane jScrollPane3;
227 private javax.swing.JLabel nombre;

```

```

228 // End of variables declaration//GEN-END:variables
229 }

```

## 2.5. Paquete Simulador

### 2.5.1. CadEntrada.java

```

1 //SimAS / Simulador
2 // Cadena de entrada
3
4 package es.uco.simas.simulador;
5
6 import javax.swing.DefaultListModel;
7 import javax.swing.JOptionPane;
8 import java.util.ArrayList;
9 import es.uco.simas.util.gramatica.*;
10
11 /**
12  * @author Vanesa
13  */
14 public class CadEntrada extends javax.swing.JFrame {
15
16     NuevaSimulacionDesc nuevo;
17     NuevaSimulacionAsc nuevoAsc;
18
19     public CadEntrada(NuevaSimulacionDesc nuevo) {
20         initComponents();
21         this.nuevo = nuevo;
22         this.jTextField1.setText("");
23         ArrayList<Terminal> cadenaEntrada = this.nuevo.ventanaPadre.
            getCadenaEntrada();
24         String aux = "";
25         int i=0;
26         DefaultListModel terminales = new DefaultListModel();
27         terminales = this.nuevo.gramatica.getTerminales();
28         this.jList1.setModel(terminales);
29
30         if(cadenaEntrada != null){
31             while (i < cadenaEntrada.size()){
32                 aux = aux+cadenaEntrada.get(i).getNombre()+" ";
33                 i++;
34             }
35         }
36         this.jTextField1.setText(aux);
37     }
38
39     public CadEntrada(NuevaSimulacionAsc nuevo) {
40         initComponents();

```

```

41     this.nuevoAsc = nuevo;
42     this.jTextField1.setText("");
43     ArrayList<Terminal> cadenaEntrada = this.nuevoAsc.ventanaPadre.
        getCadenaEntrada();
44     String aux = "";
45     int i=0;
46     DefaultListModel terminales = new DefaultListModel();
47     terminales = this.nuevoAsc.gramatica.getTerminales();
48     this.jList1.setModel(terminales);
49
50     if(cadenaEntrada != null){
51         while (i < cadenaEntrada.size()){
52             aux = aux+cadenaEntrada.get(i).getNombre()+" ";
53             i++;
54         }
55     }
56     this.jTextField1.setText(aux);
57 }
58
59 /**
60  * This method is called from within the constructor to initialize
        the form.
61  * WARNING: Do NOT modify this code. The content of this method is
        always
62  * regenerated by the Form Editor.
63  */
64 @SuppressWarnings("unchecked")
65 // <editor-fold defaultstate="collapsed" desc="Generated Code">//
        GEN-BEGIN: initComponents
66 private void initComponents() {
67
68     jPanel1 = new javax.swing.JPanel();
69     jLabel1 = new javax.swing.JLabel();
70     jTextField1 = new javax.swing.JTextField();
71     jButton1 = new javax.swing.JButton();
72     jScrollPane1 = new javax.swing.JScrollPane();
73     jList1 = new javax.swing.JList();
74     jLabel2 = new javax.swing.JLabel();
75     jButton2 = new javax.swing.JButton();
76     jButton3 = new javax.swing.JButton();
77
78     setDefaultCloseOperation(javax.swing.WindowConstants.
        DISPOSE_ON_CLOSE);
79     setTitle("Cadena de Entrada");
80     setBackground(new java.awt.Color(233, 244, 244));
81
82     jPanel1.setBackground(new java.awt.Color(233, 244, 244));
83
84     jLabel1.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
85     jLabel1.setText("Cadena de Entrada");
86
87     jButton1.setText("Borrar");

```



```

128         .addGroup(jPanel1Layout.createParallelGroup(
129             javax.swing.GroupLayout.Alignment.TRAILING)
130             .addComponent(jLabel2)
131             .addComponent(jTextField1, javax.swing.
132                 GroupLayout.PREFERRED_SIZE, 160, javax.
133                 swing.GroupLayout.PREFERRED_SIZE))
134         .addPreferredGap(javax.swing.LayoutStyle.
135             ComponentPlacement.RELATED)
136         .addComponent(jButton1)
137         .addGap(34, 34, 34))
138     .addContainerGap(34, Short.MAX_VALUE))
139 .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
140     jPanel1Layout.createSequentialGroup())
141     .addGap(0, 0, Short.MAX_VALUE)
142     .addComponent(jLabel1)
143     .addGap(105, 105, 105))
144 .addGroup(jPanel1Layout.createSequentialGroup())
145     .addGap(21, 21, 21)
146     .addComponent(jButton2)
147     .addPreferredGap(javax.swing.LayoutStyle.
148         ComponentPlacement.RELATED, javax.swing.GroupLayout
149         .DEFAULT_SIZE, Short.MAX_VALUE)
150     .addComponent(jButton3)
151     .addGap(57, 57, 57))
152 );
153 jPanel1Layout.setVerticalGroup(
154     jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
155         Alignment.LEADING)
156     .addGroup(jPanel1Layout.createSequentialGroup())
157         .addGap(19, 19, 19)
158         .addComponent(jLabel1)
159         .addGap(27, 27, 27)
160         .addGroup(jPanel1Layout.createParallelGroup(javax.swing
161             .GroupLayout.Alignment.BASELINE)
162             .addComponent(jTextField1, javax.swing.GroupLayout.
163                 PREFERRED_SIZE, javax.swing.GroupLayout.
164                 DEFAULT_SIZE, javax.swing.GroupLayout.
165                 PREFERRED_SIZE)
166             .addComponent(jButton1))
167         .addGap(26, 26, 26)
168         .addComponent(jLabel2)
169         .addPreferredGap(javax.swing.LayoutStyle.
170             ComponentPlacement.UNRELATED)
171         .addComponent(jScrollPane1, javax.swing.GroupLayout.
172             PREFERRED_SIZE, javax.swing.GroupLayout.
173             DEFAULT_SIZE, javax.swing.GroupLayout.
174             PREFERRED_SIZE)
175         .addGap(18, 18, 18)
176         .addGroup(jPanel1Layout.createParallelGroup(javax.swing
177             .GroupLayout.Alignment.BASELINE)
178             .addComponent(jButton3)
179             .addComponent(jButton2))
180         .addContainerGap(36, Short.MAX_VALUE))

```

```

164         );
165
166         javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
167             getContentPane());
168         getContentPane().setLayout(layout);
169         layout.setHorizontalGroup(
170             layout.createParallelGroup(javax.swing.GroupLayout.
171                 Alignment.LEADING)
172                 .addComponent(jPanel1, javax.swing.GroupLayout.
173                     PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
174                     javax.swing.GroupLayout.PREFERRED_SIZE)
175         );
176
177         pack();
178     } // </editor-fold> //GEN-END: initComponents
179
180     private void jButton2ActionPerformed(java.awt.event.ActionEvent evt
181     ) { //GEN-FIRST: event_jButton2ActionPerformed
182         int conf = JOptionPane.showConfirmDialog(null, "¿Desea salir?
183             ", "Salir", JOptionPane.YES_NO_OPTION);
184
185         if (conf == 0)
186             this.dispose();
187     } //GEN-LAST: event_jButton2ActionPerformed
188
189     private void jButton3ActionPerformed(java.awt.event.ActionEvent evt
190     ) { //GEN-FIRST: event_jButton3ActionPerformed
191         String cadena = this.jTextField1.getText();
192         String [] separado = cadena.split(" ");
193         ArrayList<Terminal> cadenaEntrada = new ArrayList<>();
194         int i = 0;
195         while (i < separado.length) {
196             cadenaEntrada.add(new Terminal(separado[i], separado[i]));
197             i++;
198         }
199         if (this.nuevo != null) {
200             this.nuevo.ventanaPadre.setCadenaEntrada(cadenaEntrada);
201             this.nuevo.actualizarVisualizacion();
202         }
203         if (this.nuevoAsc != null) {
204             this.nuevoAsc.ventanaPadre.setCadenaEntrada(cadenaEntrada);
205             this.nuevoAsc.actualizarVisualizacion();
206         }
207
208         this.dispose();
209     } //GEN-LAST: event_jButton3ActionPerformed

```

```

207
208     private void addTerminal(java.awt.event.MouseEvent evt) {//GEN-FIRST:event-addTerminal
209
210         StringBuilder string = new StringBuilder();
211         this.jTextField1.setText(string.append(this.jTextField1.getText()
212             .append(this.jList1.getSelectedValue()).append(" ")).
213             toString());
214     }//GEN-LAST:event-addTerminal
215
216     private void jButton1ActionPerformed(java.awt.event.ActionEvent evt
217     ) {//GEN-FIRST:event_jButton1ActionPerformed
218         String cadena = this.jTextField1.getText();
219         String [] separado = cadena.split(" ");
220         String aux = "";
221         int i = 0;
222         while(i < (separado.length -1)){
223             aux = aux + separado[i]+" ";
224             i++;
225         }
226         this.jTextField1.setText(aux);
227     }//GEN-LAST:event_jButton1ActionPerformed
228
229     /**
230      * @param args the command line arguments
231      */
232     public static void main(String args[]) {
233         /* Set the Nimbus look and feel */
234         //<editor-fold defaultstate="collapsed" desc=" Look and feel
235         setting code (optional) ">
236         /* If Nimbus (introduced in Java SE 6) is not available, stay
237         with the default look and feel.
238         * For details see http://download.oracle.com/javase/tutorial/
239         uiswing/lookandfeel/plaf.html
240         */
241         try {
242             for (javax.swing.UIManager.LookAndFeelInfo info : javax.
243                 swing.UIManager.getInstalledLookAndFeels()) {
244                 if ("Nimbus".equals(info.getName())) {
245                     javax.swing.UIManager.setLookAndFeel(info.
246                         getClassName());
247                     break;
248                 }
249             }
250         } catch (ClassNotFoundException ex) {
251             java.util.logging.Logger.getLogger(CadEntrada.class.getName()
252                 ().log(java.util.logging.Level.SEVERE, null, ex);
253         } catch (InstantiationException ex) {
254             java.util.logging.Logger.getLogger(CadEntrada.class.getName()
255                 ().log(java.util.logging.Level.SEVERE, null, ex);
256         } catch (IllegalAccessException ex) {

```

```

248         java.util.logging.Logger.getLogger(CadEntrada.class.getName
249            ()).log(java.util.logging.Level.SEVERE, null, ex);
250     } catch (javax.swing.UnsupportedLookAndFeelException ex) {
251         java.util.logging.Logger.getLogger(CadEntrada.class.getName
252            ()).log(java.util.logging.Level.SEVERE, null, ex);
253     }
254     //</editor-fold>
255     /* Create and display the form */
256     java.awt.EventQueue.invokeLater(new Runnable() {
257         public void run() {
258             // new CadEntrada().setVisible(true);
259         }
260     });
261
262     // Variables declaration - do not modify//GEN-BEGIN:variables
263     private javax.swing.JButton jButton1;
264     private javax.swing.JButton jButton2;
265     private javax.swing.JButton jButton3;
266     private javax.swing.JLabel jLabel1;
267     private javax.swing.JLabel jLabel2;
268     private javax.swing.JList jList1;
269     private javax.swing.JPanel jPanel1;
270     private javax.swing.JScrollPane jScrollPane1;
271     private javax.swing.JTextField jTextField1;
272     // End of variables declaration//GEN-END:variables
273 }

```

### 2.5.2. MiRender.java

```

1 //SimAS / Simulador
2 //Mi Render
3
4 package es.uco.simas.simulador;
5
6 import java.awt.Color;
7 import java.awt.Component;
8 import javax.swing.JTable;
9 import javax.swing.table.DefaultTableCellRenderer;
10
11 /**
12  * @author Vanesa
13  */
14 public class MiRender extends DefaultTableCellRenderer{
15
16     public Component getTableCellRendererComponent(JTable table ,
17         Object value ,
18         boolean isSelected ,
19         boolean hasFocus ,

```



```

20     int row,
21     int column
22 )
23 {
24     Component cell = super.getTableCellRendererComponent ( table ,
25         value , isSelected , hasFocus , row , column );
26     if ( value != null && column != 0 && value.toString().startsWith(
27         "E" ) && !value.toString().startsWith("Emp" ) ) {
28         cell.setForeground( Color.RED );
29     } else {
30         cell.setForeground( Color.BLACK );
31     }
32     if ( value != null ) {
33         if ( value == "Error" || value.toString().contains("conf" ) )
34             cell.setForeground( Color.RED );
35         if ( value == "Emparejar" )
36             cell.setForeground( Color.BLUE );
37         if ( value == "Aceptar" )
38             cell.setForeground( Color.GREEN );
39         if ( value != null && column != 0 && value.toString().startsWith(
40             "d" ) )
41             cell.setForeground( Color.BLUE );
42         if ( value != null && column != 0 && value.toString().startsWith(
43             "r" ) )
44             cell.setForeground( Color.MAGENTA );
45         if ( value != null && column != 0 && value.toString().contains("
46             *" ) )
47             cell.setForeground( Color.GRAY );
48     }
49     return cell;
50 }

```

### 2.5.3. NuevaFuncionError.java

```

1  //SimAS / Simulador
2  // Nueva Funcion error
3
4  package es.uco.simas.simulador;
5
6  import es.uco.simas.editor.FuncionError;
7  import es.uco.simas.editor.ParteAccion;
8  import es.uco.simas.editor.TablaLR;
9  import es.uco.simas.editor.TablaPredictiva;
10 import es.uco.simas.util.gramatica.Gramatica;
11 import es.uco.simas.util.gramatica.Terminal;
12 import java.util.ArrayList;
13 import javax.swing.DefaultComboBoxModel;
14 import javax.swing.DefaultListModel;

```

```

15 import javax.swing.JOptionPane;
16
17 /**
18  * @author vanesa
19  */
20 public class NuevaFuncionError extends javax.swing.JFrame {
21
22     Gramatica gramatica;
23     TablaPredictiva tPredictiva;
24     TablaLR tlr;
25     PanelNuevaSimDescPaso4 paso4;
26     PanelNuevaSimAscPaso5 paso5;
27     int sim = 0;
28
29     public NuevaFuncionError(Gramatica gramatica,
30         PanelNuevaSimDescPaso4 paso4) {
31         this.gramatica = gramatica;
32         initComponents();
33         inicializarCombos();
34         this.tPredictiva = this.gramatica.getTPredictiva();
35         this.paso4 = paso4;
36         this.sim = 1;
37     }
38
39     public NuevaFuncionError(Gramatica gramatica,
40         PanelNuevaSimAscPaso5 paso5, int i) {
41         this.gramatica = gramatica;
42         initComponents();
43         inicializarCombos();
44         this.tlr = this.gramatica.getTlr();
45         this.paso5 = paso5;
46         this.sim = 2;
47     }
48
49     /**
50      * This method is called from within the constructor to initialize
51      * the form.
52      * WARNING: Do NOT modify this code. The content of this method is
53      * always
54      * regenerated by the Form Editor.
55      */
56     @SuppressWarnings("unchecked")
57     // <editor-fold defaultstate="collapsed" desc="Generated Code">
58     GEN-BEGIN: initComponents
59     private void initComponents() {
60
61         jPanel1 = new javax.swing.JPanel();
62         jLabel1 = new javax.swing.JLabel();
63         jLabel2 = new javax.swing.JLabel();
64         jLabel3 = new javax.swing.JLabel();
65         jLabel4 = new javax.swing.JLabel();
66         jTextField1 = new javax.swing.JTextField();
67         jLabel5 = new javax.swing.JLabel();
68         jComboBox1 = new javax.swing.JComboBox();

```

```

63     jLabel5 = new javax.swing.JLabel();
64     jComboBox2 = new javax.swing.JComboBox();
65     jLabel6 = new javax.swing.JLabel();
66     jTextField2 = new javax.swing.JTextField();
67     jButton1 = new javax.swing.JButton();
68     jButton2 = new javax.swing.JButton();
69
70     setDefaultCloseOperation(javax.swing.WindowConstants.
71         EXIT_ON_CLOSE);
72     setTitle("Nueva Funcion de error");
73
74     jPanel1.setBackground(new java.awt.Color(233, 244, 244));
75
76     jLabel1.setFont(new java.awt.Font("Ubuntu", 0, 18)); // NOI18N
77     jLabel1.setText("Nueva Funcion Error");
78
79     jLabel2.setText("Identificador");
80
81     jLabel3.setText("E");
82
83     jLabel4.setText("Accion");
84
85     jComboBox1.setFont(new java.awt.Font("Ubuntu", 0, 14)); //
86         NOI18N
87     jComboBox1.setModel(new javax.swing.DefaultComboBoxModel(new
88         String[] { "1. Insertar un S mbolo en la Entrada", "2.
89         Borrar un S mbolo de la Entrada", "3. Modificar un
90         S mbolo de la Entrada", "4. Insertar un S mbolo de la
91         Pila", "5. Borrar un S mbolo de la Pila", "6. Modificar un
92         S mbolo de la Pila", "7. Terminar el an lisis" }));
93     jComboBox1.addItemListener(new java.awt.event.ItemListener() {
94         public void itemStateChanged(java.awt.event.ItemEvent evt)
95         {
96             combo(evt);
97         }
98     });
99
100     jLabel5.setText("S mbolo");
101
102     jLabel6.setText("Mensaje");
103
104     jButton1.setText("Cancelar");
105     jButton1.addActionListener(new java.awt.event.ActionListener()
106     {
107         public void actionPerformed(java.awt.event.ActionEvent evt)
108         {
109             jButton1ActionPerformed(evt);
110         }
111     });
112
113     jButton2.setText("Aceptar");
114     jButton2.addActionListener(new java.awt.event.ActionListener()
115     {

```

```

105         public void actionPerformed(java.awt.event.ActionEvent evt)
106         {
107             jButton2ActionPerformed(evt);
108         }
109     });
110     javax.swing.GroupLayout jPanel1Layout = new javax.swing.
111         GroupLayout(jPanel1);
112     jPanel1.setLayout(jPanel1Layout);
113     jPanel1Layout.setHorizontalGroup(
114         jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
115             Alignment.LEADING)
116             .addGroup(jPanel1Layout.createSequentialGroup()
117                 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.
118                     GroupLayout.Alignment.TRAILING)
119                     .addGroup(jPanel1Layout.
120                         createSequentialGroup()
121                         .addComponent(jLabel2)
122                         .addGap(43, 43, 43)
123                         .addGroup(jPanel1Layout.
124                             createParallelGroup(javax.swing.
125                                 GroupLayout.Alignment.LEADING)
126                                 .addComponent(jLabel1)
127                                 .addGroup(jPanel1Layout.
128                                     createSequentialGroup()
129                                     .addComponent(jLabel3)
130                                     .addGap(4, 4, 4)
131                                     .addComponent(jTextField1,
132                                         javax.swing.GroupLayout.
133                                             PREFERRED_SIZE, 24, javax.
134                                             swing.GroupLayout.
135                                             PREFERRED_SIZE)
136                                     .addGap(77, 77, 77)
137                                     .addComponent(jLabel6))))
138                 .addComponent(jButton1, javax.swing.
139                     GroupLayout.Alignment.LEADING))
140             .addPreferredGap(javax.swing.LayoutStyle.
141                 ComponentPlacement.RELATED, javax.swing.
142                 GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
143             .addComponent(jButton2))
144         .addGroup(jPanel1Layout.createSequentialGroup()
145             .addGroup(jPanel1Layout.createParallelGroup(javax.swing.
146                 GroupLayout.Alignment.LEADING)
147                 .addGroup(javax.swing.GroupLayout.Alignment.
148                     TRAILING, jPanel1Layout.
149                     createSequentialGroup()
150                     .addGap(0, 0, Short.MAX_VALUE)
151                     .addComponent(jLabel4)
152                     .addGap(209, 209, 209))

```

```

139         .addGroup(jPanel1Layout .
140             createSequentialGroup ()
141             .addGroup(jPanel1Layout .
142                 createParallelGroup (javax.swing .
143                     GroupLayout.Alignment.LEADING)
144                     .addComponent(jComboBox2, javax .
145                         swing.GroupLayout .
146                             PREFERRED.SIZE, 215, javax .
147                                 swing.GroupLayout .
148                                     PREFERRED.SIZE)
149                     .addComponent(jComboBox1, javax .
150                         swing.GroupLayout .
151                             PREFERRED.SIZE, 246, javax .
152                                 swing.GroupLayout .
153                                     PREFERRED.SIZE)
154                     .addComponent(jLabel5))
155             .addPreferredGap (javax.swing .
156                 LayoutStyle.ComponentPlacement .
157                     RELATED, javax.swing.GroupLayout .
158                         DEFAULT.SIZE, Short.MAX.VALUE)))
159         .addComponent(jTextField2, javax.swing .
160             GroupLayout.PREFERRED.SIZE, 221, javax .
161                 swing.GroupLayout.PREFERRED.SIZE)))
162         .addGap(30, 30, 30))
163     );
164     jPanel1Layout.setVerticalGroup(
165         jPanel1Layout.createParallelGroup (javax.swing.GroupLayout .
166             Alignment.LEADING)
167         .addGroup(jPanel1Layout.createSequentialGroup ()
168             .addGap(20, 20, 20)
169             .addComponent(jLabel1)
170             .addGap(23, 23, 23)
171             .addGroup(jPanel1Layout.createParallelGroup (javax.swing
172                 .GroupLayout.Alignment.BASELINE)
173                 .addComponent(jLabel2)
174                 .addComponent(jLabel3)
175                 .addComponent(jTextField1, javax.swing.GroupLayout .
176                     PREFERRED.SIZE, javax.swing.GroupLayout .
177                         DEFAULT.SIZE, javax.swing.GroupLayout .
178                             PREFERRED.SIZE)
179                 .addComponent(jLabel6))
180             .addGap(18, 18, 18)
181             .addGroup(jPanel1Layout.createParallelGroup (javax.swing
182                 .GroupLayout.Alignment.LEADING)
183                 .addGroup(jPanel1Layout.createSequentialGroup ()
184                     .addComponent(jLabel4)
185                     .addPreferredGap (javax.swing.LayoutStyle .
186                         ComponentPlacement.UNRELATED)
187                     .addComponent(jComboBox1, javax.swing .
188                         GroupLayout.PREFERRED.SIZE, javax.swing .
189                             GroupLayout.DEFAULT.SIZE, javax.swing .
190                                 GroupLayout.PREFERRED.SIZE)
191                     .addGap(62, 62, 62)

```

```

166         .addComponent(jLabel5)
167         .addPreferredGap(javax.swing.LayoutStyle.
            ComponentPlacement.RELATED)
168         .addComponent(jComboBox2, javax.swing.
            GroupLayout.PREFERRED_SIZE, javax.swing.
            GroupLayout.DEFAULT_SIZE, javax.swing.
            GroupLayout.PREFERRED_SIZE))
169         .addComponent(jTextField2, javax.swing.GroupLayout.
            PREFERRED_SIZE, 180, javax.swing.GroupLayout.
            PREFERRED_SIZE))
170         .addPreferredGap(javax.swing.LayoutStyle.
            ComponentPlacement.RELATED, 45, Short.MAX_VALUE)
171         .addGroup(jPanel1Layout.createParallelGroup(javax.swing.
            GroupLayout.Alignment.BASELINE)
172             .addComponent(jButton1)
173             .addComponent(jButton2))
174         .addGap(24, 24, 24))
175     );
176
177     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
        getContentPane());
178     getContentPane().setLayout(layout);
179     layout.setHorizontalGroup(
180         layout.createParallelGroup(javax.swing.GroupLayout.
            Alignment.LEADING)
181         .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE
            , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE
            )
182     );
183     layout.setVerticalGroup(
184         layout.createParallelGroup(javax.swing.GroupLayout.
            Alignment.LEADING)
185         .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE
            , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE
            )
186     );
187
188     pack();
189 } // </editor-fold> //GEN-END: initComponents
190
191
192 private void inicializarCombos() {
193     ArrayList<String> listaCombo = new ArrayList();
194     DefaultListModel terminales= new DefaultListModel();
195     int i= 0;
196     terminales = this.gramatica.getTerminales();
197
198     while(true) {
199         if(i >= terminales.size()) {
200             break;
201         }
202         if(i < terminales.size()) {
203             listaCombo.add((String) terminales.get(i));

```

```

204         i++;
205     }
206 }
207 DefaultComboBoxModel combo = new DefaultComboBoxModel(
208     listaCombo.toArray());
209 this.jComboBox2.setModel(combo);
210 }
211
212 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt
213 ) { //GEN-FIRST:event_jButton1ActionPerformed
214     int conf = JOptionPane.showConfirmDialog(null, "¿Desea salir
215     ?" , " Salir" ,JOptionPane.YES_NO_OPTION);
216
217     if (conf==0)
218         this.dispose();
219 } //GEN-LAST:event_jButton1ActionPerformed
220
221 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt
222 ) { //GEN-FIRST:event_jButton2ActionPerformed
223     ArrayList<FuncionError> funciones = new ArrayList();
224     if (this.sim ==1)
225         funciones = this.gramatica.getTPredictiva().getFunError();
226     if (this.sim ==2)
227         funciones = this.gramatica.getTlr().getTAccion().
228         getFunError();
229
230     int i=0;
231     if ("".equals(this.jTextField1.getText().toString())){
232         JOptionPane.showConfirmDialog(null, "El identificador no
233         puede estar vacío." , " Salir" ,JOptionPane.CLOSED_OPTION
234         );
235     }else{
236         int j=0;
237         while(i < funciones.size()){
238             if(funciones.get(i).getIdentificador()==Integer.
239             parseInt(this.jTextField1.getText())){
240                 JOptionPane.showConfirmDialog(null, "El
241                 identificador está repetido." , " Salir" ,
242                 JOptionPane.CLOSED_OPTION);
243                 j=1;
244                 break;
245             }else
246                 i++;
247         }
248
249         if(j==0) {
250             int id = Integer.parseInt(this.jTextField1.getText().
251             toString());
252             int accion = this.jComboBox1.getSelectedIndex()+1;
253             String mensaje = this.jTextField2.toString();
254             Terminal term = new Terminal(this.jComboBox2.
255             getSelectedItem().toString(),this.jComboBox2.

```

```

245         getSelectedItem().toString());
246         FuncionError funError = new FuncionError(id, accion,
247             mensaje);
248         funError.setSimbolo(term);
249         if (sim==1){
250             this.tPredictiva.crearFunError(funError);
251             this.gramatica.setTPredictiva(tPredictiva);
252             if (this.paso4 != null)
253                 this.paso4.funcionError();
254         }
255         if (sim==2){
256             ParteAccion ac = this.tlr.getTAccion();
257             ac.crearFunError(funError);
258             this.tlr.setTAccion(ac);
259             this.gramatica.setTlr(this.tlr);
260             if (this.paso5 != null)
261                 this.paso5.funcionError();
262         }
263         this.dispose();
264     }
265 } //GEN-LAST: event_jButton2ActionPerformed
266
267 private void combo(java.awt.event.ItemEvent evt) { //GEN-FIRST:
268     event_combo
269     if (this.jComboBox1.getSelectedIndex() ==1 || this.jComboBox1.
270         getSelectedIndex() ==4 || this.jComboBox1.getSelectedIndex
271         ()==6){
272         this.jLabel5.setVisible(false);
273         this.jComboBox2.setVisible(false);
274     } else {
275         this.jLabel5.setVisible(true);
276         this.jComboBox2.setVisible(true);
277     }
278 } //GEN-LAST: event_combo
279
280 /**
281  * @param args the command line arguments
282  */
283 public static void main(String args[]) {
284     /* Set the Nimbus look and feel */
285     //<editor-fold defaultstate="collapsed" desc=" Look and feel
286     setting code (optional) ">
287     /* If Nimbus (introduced in Java SE 6) is not available, stay
288     with the default look and feel.
289     * For details see http://download.oracle.com/javase/tutorial/
290     uiswing/lookandfeel/plaf.html
291     */
292     try {
293         for (javax.swing.UIManager.LookAndFeelInfo info : javax.
294             swing.UIManager.getInstalledLookAndFeels()) {
295             if ("Nimbus".equals(info.getName())) {

```



```

289         javax.swing.UIManager.setLookAndFeel(info .
290             getClassName());
291         break;
292     }
293 } catch (ClassNotFoundException ex) {
294     java.util.logging.Logger.getLogger(NuevaFuncionError.class .
295         getName()).log(java.util.logging.Level.SEVERE, null, ex
296     );
297 } catch (InstantiationException ex) {
298     java.util.logging.Logger.getLogger(NuevaFuncionError.class .
299         getName()).log(java.util.logging.Level.SEVERE, null, ex
300     );
301 } catch (IllegalAccessException ex) {
302     java.util.logging.Logger.getLogger(NuevaFuncionError.class .
303         getName()).log(java.util.logging.Level.SEVERE, null, ex
304     );
305 } catch (javax.swing.UnsupportedLookAndFeelException ex) {
306     java.util.logging.Logger.getLogger(NuevaFuncionError.class .
307         getName()).log(java.util.logging.Level.SEVERE, null, ex
308     );
309 }
310 //</editor-fold>
311
312 /* Create and display the form */
313 java.awt.EventQueue.invokeLater(new Runnable() {
314     public void run() {
315         //new NuevaFuncionError().setVisible(true);
316     }
317 });
318
319 // Variables declaration - do not modify//GEN-BEGIN:variables
320 private javax.swing.JButton jButton1;
321 private javax.swing.JButton jButton2;
322 private javax.swing.JComboBox jComboBox1;
323 private javax.swing.JComboBox jComboBox2;
324 private javax.swing.JLabel jLabel1;
325 private javax.swing.JLabel jLabel2;
326 private javax.swing.JLabel jLabel3;
327 private javax.swing.JLabel jLabel4;
328 private javax.swing.JLabel jLabel5;
329 private javax.swing.JLabel jLabel6;
330 private javax.swing.JPanel jPanel1;
331 private javax.swing.JTextField jTextField1;
332 private javax.swing.JTextField jTextField2;
333 // End of variables declaration//GEN-END:variables
334 }

```

#### 2.5.4. NuevaSimulacionAsc.java

```

1 //SimAS / Simulador
2 //Nueva Simulacion Ascendente
3
4 package es.uco.simas.simulador;
5
6 import com.itextpdf.text.DocumentException;
7 import es.uco.simas.editor.Editor;
8 import es.uco.simas.editor.FuncionError;
9 import es.uco.simas.util.gramatica.Gramatica;
10 import es.uco.simas.util.gramatica.Produccion;
11 import es.uco.simas.util.gramatica.Terminal;
12 import java.io.File;
13 import java.util.ArrayList;
14 import java.util.Stack;
15 import java.util.logging.Level;
16 import java.util.logging.Logger;
17 import javax.swing.JFileChooser;
18 import javax.swing.JOptionPane;
19 import javax.swing.filechooser.FileNameExtensionFilter;
20 import javax.swing.table.DefaultTableModel;
21
22 /**
23  * @author vanesa
24  */
25 public class NuevaSimulacionAsc extends javax.swing.JFrame {
26
27     public Gramatica gramatica;
28     public Simulador ventanaPadre;
29     private DefaultTableModel tabla = new DefaultTableModel();
30     Stack<String> pila = new Stack<String>();
31     Stack<String> entrada = new Stack<String>();
32     private String accion = "";
33
34     public NuevaSimulacionAsc(Gramatica gramatica, Simulador
        ventanaPadre) {
35         initComponents();
36
37         this.gramatica = gramatica;
38         this.ventanaPadre = ventanaPadre;
39         if (this.ventanaPadre.getMetodoAscendente() == 0) {
40             this.jLabel1.setText("Simulacion Ascendente SLR");
41         }
42         if (this.ventanaPadre.getMetodoAscendente() == 1) {
43             this.jLabel1.setText("Simulacion Ascendente LR-canonico");
44         }
45         if (this.ventanaPadre.getMetodoAscendente() == 3) {
46             this.jLabel2.setText("Simulacion Ascendente LALR");
47         }
48
49         if (this.ventanaPadre.getCadenaEntrada() != null) {
50             iniciarSimulacion();
51         }
52         this.jTableSim.setModel(this.tabla);

```

```
53         this.jTableSim.setDefaultRenderer (Object.class , new MiRender()  
54     );  
55  
56     public void actualizarVisualizacion() {  
57         String cadena ="";  
58         this.pila.removeAllElements();  
59         this.entrada.removeAllElements();  
60         this.tabla = new DefaultTableModel();  
61         this.jTableSim.setModel(this.tabla);  
62  
63         ArrayList<Terminal> cadenaEntrada = this.ventanaPadre.  
64             getCadenaEntrada();  
65         int i = 0;  
66         while (i < cadenaEntrada.size()) {  
67             cadena = cadena+cadenaEntrada.get(i).getNombre()+" ";  
68             i++;  
69         }  
70         this.jTextField1.setText(cadena);  
71         if (!cadena.equals("")) {  
72             this.iniciarSimulacion();  
73         }  
74     }  
75  
76     private void iniciarSimulacion() {  
77         this.tabla.addColumn(" Pila");  
78         this.tabla.addColumn(" Entrada");  
79         this.tabla.addColumn(" Accion");  
80  
81         ArrayList<Terminal> cadEntrada = this.ventanaPadre.  
82             getCadenaEntrada();  
83         int i=0;  
84         this.entrada.push("$");  
85         i = cadEntrada.size() -1;  
86         while(i >= 0) {  
87             this.entrada.push(cadEntrada.get(i).getNombre());  
88             i--;  
89         }  
90         this.pila.push("0");  
91  
92         this.accion = this.buscarTablaAccion(0, this.entrada.peek());  
93  
94         Object [] linea = new Object[] {  
95             mostrarPila(this.pila),  
96             mostrarEntrada(this.entrada),  
97             accion ,  
98         };  
99         this.tabla.addRow(linea);  
100     }  
101  
102     private String mostrarPila(Stack pila) {  
103         int i = 0;
```

```

103     String str = "";
104     while(i < pila.size()){
105         str = str + " "+pila.get(i);
106         i++;
107     }
108     return str;
109 }
110
111 private String mostrarEntrada(Stack pila){
112     int i = pila.size()-1;
113     String str = "";
114     while(i >= 0){
115         str = str + " "+pila.get(i);
116         i--;
117     }
118     return str;
119 }
120
121 private String buscarTablaAccion(int fila, String columna){
122     DefaultTableModel tablaAccion = this.gramatica.getTlr().
        getTAccion().getMatrizAccion();
123     int i=0;
124     int fil = -1;
125     int col = -1;
126     Object celda = "";
127
128     while(i < tablaAccion.getRowCount()){
129         if(tablaAccion.getValueAt(i, 0).toString().contains("-")){
130             int a=tablaAccion.getValueAt(i, 0).toString().indexOf(
                "-");
131             int numero = Integer.parseInt(tablaAccion.getValueAt(i,
                0).toString().substring(0, a));
132             if(fila == numero){
133                 fil = i;
134                 break;
135             }else{
136                 i++;
137             }
138         }else{
139             if(fila == Integer.parseInt(tablaAccion.getValueAt(i,
                0).toString())){
140                 fil = i;
141                 break;
142             }else{
143                 i++;
144             }
145         }
146     }
147     i = 1;
148     while(i < tablaAccion.getColumnCount()){
149         if(columna.equals(tablaAccion.getColumnName(i))){
150             col = i;
151             break;

```

```

152         }else{
153             i++;
154         }
155     }
156     if( fil != -1 && col != -1){
157         celda = tablaAccion.getValueAt( fil , col);
158     }
159     if( celda == null){
160         return "Error";
161     }else{
162         return celda.toString();
163     }
164 }
165
166 private String buscarTablaIra(int fila , String columna){
167     DefaultTableModel tablaIra = this.gramatica.getTlr().getTlrA().
        getMatrizIrA();
168     DefaultTableModel tablaAccion = this.gramatica.getTlr().
        getTAccion().getMatrizAccion();
169     int i=0;
170     int fil = -1;
171     int col = -1;
172     Object celda = "";
173
174     while(i < tablaIra.getRowCount()){
175         if(tablaAccion.getValueAt(i, 0).toString().contains("-")){
176             int a = tablaAccion.getValueAt(i, 0).toString().indexOf
                ("-");
177             int numero = Integer.parseInt(tablaAccion.getValueAt(i,
                0).toString().substring(0, a));
178             if( fila == numero){
179                 fil = i;
180                 break;
181             }else{
182                 i++;
183             }
184         }else{
185             if( fila == Integer.parseInt(tablaAccion.getValueAt(i,
                0).toString())){
186                 fil = i;
187                 break;
188             }else{
189                 i++;
190             }
191         }
192     }
193     i = 0;
194     while(i < tablaIra.getColumnCount()){
195         if( columna.equals( tablaIra.getColumnName(i))){
196             col = i;
197             break;
198         }else{
199             i++;

```

```

200     }
201 }
202 if (fil != -1 && col != -1){
203     celda = tablaIra.getValueAt(fil , col);
204 }
205 if (celda == null){
206     return "Error";
207 }else{
208     return celda.toString();
209 }
210 }
211
212 private void siguientePaso(){
213     int desp = -1;
214     int red = -1;
215     int i=0;
216     ArrayList<Produccion> prod = this.gramatica.getPr();
217     if (this.accion.equals("Aceptar") || this.accion.equals("Error")
218         || this.accion.contains("conf")){
219         this.jButton4.setEnabled(false);
220     }else{
221         if (this.accion.startsWith("d")){
222             this.pila.push(this.entrada.peek());
223             this.entrada.pop();
224             this.pila.push(this.accion.substring(1, this.accion.
225                 length()));
226         }
227         if (this.accion.startsWith("r")){
228             int aux;
229
230             if (this.accion.substring(2, 2)==" "){
231                 aux=1;
232             }else{
233                 aux=2;
234             }
235
236             red= Integer.parseInt(this.accion.substring(1, aux));
237             Produccion pr = prod.get(red);
238
239             if (pr.getConsec().get(0).getNombre().equals("\u03b5")){
240                 this.pila.push(pr.getAntec().getSimboloNT().
241                     getNombre());
242                 String num =(this.buscarTablaIra(Integer.parseInt(
243                     this.pila.get(this.pila.size()-2), this.pila.
244                     get(this.pila.size()-1)));
245                 this.pila.push(num);
246             }else{
247                 this.pila.pop();
248                 i=pr.getConsec().size()-1;
249                 while(i >= 0){
250                     if (pr.getConsec().get(i).getNombre().equals(
251                         this.pila.peek())){
252                         this.pila.pop();

```

```

247         if(i > 0)
248             this.pila.pop();
249     }
250     i--;
251 }
252 this.pila.push(pr.getAntec().getSimboloNT().
    getNombre());
253
254 String num= "";
255 if(this.pila.get(this.pila.size()-2).contains("-"))
256 {
257     int a=this.pila.get(this.pila.size()-2).
        indexOf("-");
258     int numero = Integer.parseInt(this.pila.get(
        this.pila.size()-2).substring(0, a));
259     num =(this.buscarTablaIra(numero, this.pila.get
        (this.pila.size()-1)));
260 }else{
261     num =(this.buscarTablaIra(Integer.parseInt(this
        .pila.get(this.pila.size()-2)), this.pila.
        get(this.pila.size()-1)));
262 }
263 this.pila.push(num);
264 }
265
266 if(this.pila.peek().contains("-")){
267     int a=this.pila.peek().indexOf("-");
268     int numero = Integer.parseInt(this.pila.peek().
        substring(0, a));
269     this.accion = this.buscarTablaAccion(numero, this.
        entrada.peek());
270 }else{
271     this.accion = this.buscarTablaAccion(Integer.parseInt(
        this.pila.peek()), this.entrada.peek());
272 }
273
274 if(this.accion.startsWith("r")){
275     Produccion pr = prod.get(Integer.parseInt(this.accion.
        substring(1, this.accion.length())));
276     this.accion = this.accion + " " + pr.getAntec().
        getSimboloNT().getNombre();
277     this.accion = this.accion + " \u2192 ";
278     i=0;
279     while(i < pr.getConsec().size()){
280         this.accion = this.accion +" "+pr.getConsec().get(i
        ).getNombre();
281         i++;
282     }
283 }
284
285 if(this.accion.startsWith("E") && !this.accion.equals("
    Error")){

```

```

286         funError();
287     }
288
289     Object [] linea = new Object[] {
290         mostrarPila(this.pila),
291         mostrarEntrada(this.entrada),
292         this.accion,
293     };
294     this.tabla.addRow(linea);
295 }
296 }
297
298 public DefaultTableModel getTabla() {
299     this jButton5ActionPerformed(null);
300     return this.tabla;
301 }
302
303 private void funError() {
304     ArrayList<FuncionError> funError = this.gramatica.getTlr().
305         getTAccion().getFunError();
306     FuncionError fun = new FuncionError();
307     int i = 0;
308     int num = Integer.parseInt(this.accion.substring(1));
309
310     while(i < funError.size()) {
311         if(funError.get(i).getIdentificador() == num) {
312             fun = funError.get(i);
313             break;
314         } else {
315             i++;
316         }
317
318         if(fun.getAccion() == 1) {
319             this.entrada.push(fun.getSimbolo().getNombre());
320         }
321         if(fun.getAccion() == 2) {
322             this.entrada.pop();
323         }
324         if(fun.getAccion() == 3) {
325             this.entrada.pop();
326             this.entrada.push(fun.getSimbolo().getNombre());
327         }
328         if(fun.getAccion() == 4) {
329             this.pila.push(fun.getSimbolo().getNombre());
330         }
331         if(fun.getAccion() == 5) {
332             this.pila.pop();
333         }
334         if(fun.getAccion() == 6) {
335             this.pila.pop();
336             this.pila.push(fun.getSimbolo().getNombre());
337         }
338         if(fun.getAccion() == 7) {

```



```

338         this.accion = "Fin";
339     }
340 }
341
342 /**
343  * This method is called from within the constructor to initialize
344  * the form.
345  * WARNING: Do NOT modify this code. The content of this method is
346  * always
347  * regenerated by the Form Editor.
348  */
349 @SuppressWarnings("unchecked")
350 // <editor-fold defaultstate="collapsed" desc="Generated Code">
351 GEN-BEGIN: initComponents
352 private void initComponents() {
353
354     jPanel1 = new javax.swing.JPanel();
355     jLabel1 = new javax.swing.JLabel();
356     jLabel2 = new javax.swing.JLabel();
357     jTextField1 = new javax.swing.JTextField();
358     jButton1 = new javax.swing.JButton();
359     jButton2 = new javax.swing.JButton();
360     jButton3 = new javax.swing.JButton();
361     jButton4 = new javax.swing.JButton();
362     jButton5 = new javax.swing.JButton();
363     jScrollPane1 = new javax.swing.JScrollPane();
364     jTableSim = new javax.swing.JTable();
365     jButton6 = new javax.swing.JButton();
366     jButton7 = new javax.swing.JButton();
367
368     setDefaultCloseOperation(javax.swing.WindowConstants.
369         DISPOSE_ON_CLOSE);
370     setTitle("Simulacion Ascendente");
371     setBackground(new java.awt.Color(233, 242, 242));
372
373     jPanel1.setBackground(new java.awt.Color(233, 242, 242));
374
375     jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
376     jLabel1.setText("Simulacion Ascendente");
377
378     jLabel2.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
379     jLabel2.setText("Cadena de Entrada: ");
380
381     jButton1.setText("Editar");
382     jButton1.addActionListener(new java.awt.event.ActionListener()
383     {
384         public void actionPerformed(java.awt.event.ActionEvent evt)
385         {
386             jButton1ActionPerformed(evt);
387         }
388     });

```

```

384     jButton2.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/primero.png"))); //
        NOI18N
385     jButton2.addActionListener(new java.awt.event.ActionListener()
        {
386         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
387             jButton2ActionPerformed(evt);
388         }
389     });
390
391     jButton3.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/anterior.png"))); //
        NOI18N
392     jButton3.addActionListener(new java.awt.event.ActionListener()
        {
393         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
394             jButton3ActionPerformed(evt);
395         }
396     });
397
398     jButton4.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/siguiente.png"))); //
        NOI18N
399     jButton4.addActionListener(new java.awt.event.ActionListener()
        {
400         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
401             jButton4ActionPerformed(evt);
402         }
403     });
404
405     jButton5.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/ultimo.png"))); //
        NOI18N
406     jButton5.addActionListener(new java.awt.event.ActionListener()
        {
407         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
408             jButton5ActionPerformed(evt);
409         }
410     });
411
412     jTableSim.setModel(new javax.swing.table.DefaultTableModel(
413         new Object [][] {
414             {null, null, null, null},
415             {null, null, null, null},
416             {null, null, null, null},
417             {null, null, null, null}
418         },
419         new String [] {
420             "Title 1", "Title 2", "Title 3", "Title 4"

```

```

421     }
422   ));
423   jScrollPane1.setViewportViewView(jTableSim);
424
425   jButton6.setText("Cancelar");
426   jButton6.addActionListener(new java.awt.event.ActionListener()
427   {
428       public void actionPerformed(java.awt.event.ActionEvent evt)
429       {
430           jButton6ActionPerformed(evt);
431       }
432   });
433
434   jButton7.setText("Informe de la Simulacion");
435   jButton7.addActionListener(new java.awt.event.ActionListener()
436   {
437       public void actionPerformed(java.awt.event.ActionEvent evt)
438       {
439           jButton7ActionPerformed(evt);
440       }
441   });
442
443   javax.swing.GroupLayout jPanel1Layout = new javax.swing.
444   GroupLayout(jPanel1);
445   jPanel1.setLayout(jPanel1Layout);
446   jPanel1Layout.setHorizontalGroup(
447       jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
448       Alignment.LEADING)
449       .addGroup(jPanel1Layout.createSequentialGroup()
450       .addGroup(jPanel1Layout.createParallelGroup(javax.swing.
451       GroupLayout.Alignment.LEADING)
452       .addGroup(jPanel1Layout.createSequentialGroup()
453       .addGap(48, 48, 48)
454       .addGroup(jPanel1Layout.createParallelGroup(javax.swing.
455       GroupLayout.Alignment.LEADING,
456       false)
457       .addGroup(jPanel1Layout.
458       createSequentialGroup()

```

```

459         .addGap(18, 18, 18)
460         .addComponent(jButton5)
461         .addGap(0, 0, Short.MAX_VALUE))
462     .addGroup(jPanel1Layout .
        createSequentialGroup ()
463         .addComponent(jTextField1 ,
            javax.swing.GroupLayout .
                PREFERRED_SIZE, 300, javax .
                    swing.GroupLayout .
                        PREFERRED_SIZE)
464         .addPreferredGap(javax.swing .
            LayoutStyle .
                ComponentPlacement.RELATED,
                javax.swing.GroupLayout .
                    DEFAULT_SIZE, Short .
                        MAX_VALUE)
465         .addComponent(jButton1)
466         .addGap(19, 19, 19))))
467     .addGroup(javax.swing.GroupLayout.Alignment
        .TRAILING, jPanel1Layout .
            createSequentialGroup ()
468         .addComponent(jButton6)
469         .addPreferredGap(javax.swing .
            LayoutStyle.ComponentPlacement .
                RELATED, javax.swing.GroupLayout .
                    DEFAULT_SIZE, Short.MAX_VALUE)
470         .addComponent(jButton7))
471     .addComponent(jScrollPane1 , javax.swing .
        GroupLayout.Alignment.TRAILING, javax .
            swing.GroupLayout.PREFERRED_SIZE, 567,
            javax.swing.GroupLayout.PREFERRED_SIZE)
        ))
472     .addGroup(jPanel1Layout.createSequentialGroup ()
473         .addGap(110, 110, 110)
474         .addComponent(jLabel1)))
475     .addContainerGap(46, Short.MAX_VALUE))
476 );
477 jPanel1Layout.setVerticalGroup(
478     jPanel1Layout.createParallelGroup(javax.swing.GroupLayout .
        Alignment.LEADING)
479     .addGroup(jPanel1Layout.createSequentialGroup ()
480         .addGap(23, 23, 23)
481         .addComponent(jLabel1)
482         .addGap(35, 35, 35)
483         .addGroup(jPanel1Layout.createParallelGroup(javax.swing
            GroupLayout.Alignment.BASELINE)
484             .addComponent(jLabel2)
485             .addComponent(jTextField1 , javax.swing.GroupLayout .
                PREFERRED_SIZE, javax.swing.GroupLayout .
                    DEFAULT_SIZE, javax.swing.GroupLayout .
                        PREFERRED_SIZE)
486             .addComponent(jButton1))
487         .addGap(30, 30, 30)

```

```

488         .addGroup(jPanel1Layout.createParallelGroup(javax.swing
489             .GroupLayout.Alignment.LEADING)
490             .addComponent(jButton2)
491             .addComponent(jButton3)
492             .addComponent(jButton4)
493             .addComponent(jButton5))
494         .addPreferredGap(javax.swing.LayoutStyle.
495             ComponentPlacement.UNRELATED)
496         .addComponent(jScrollPane1, javax.swing.GroupLayout.
497             PREFERRED_SIZE, 410, javax.swing.GroupLayout.
498             PREFERRED_SIZE)
499         .addPreferredGap(javax.swing.LayoutStyle.
500             ComponentPlacement.RELATED, 32, Short.MAX_VALUE)
501         .addGroup(jPanel1Layout.createParallelGroup(javax.swing
502             .GroupLayout.Alignment.BASELINE)
503             .addComponent(jButton6)
504             .addComponent(jButton7))
505         .addGap(20, 20, 20))
506     );
507
508     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
509         getContentPane());
510     getContentPane().setLayout(layout);
511     layout.setHorizontalGroup(
512         layout.createParallelGroup(javax.swing.GroupLayout.
513             Alignment.LEADING)
514         .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE
515             , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE
516             )
517     );
518     layout.setVerticalGroup(
519         layout.createParallelGroup(javax.swing.GroupLayout.
520             Alignment.LEADING)
521         .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE
522             , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE
523             )
524     );
525
526     pack();
527 } // </editor-fold> //GEN-END: initComponents
528
529 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt
530 ) { //GEN-FIRST: event_jButton2ActionPerformed
531     DefaultTableModel tabla2 = new DefaultTableModel();
532     this.tabla = tabla2;
533     this.jTableSim.setModel(this.tabla);
534     this.pila.removeAllElements();
535     this.entrada.removeAllElements();
536     this.accion = "";
537     this.iniciarSimulacion();
538 } //GEN-LAST: event_jButton2ActionPerformed

```

```

526 private void jButton3ActionPerformed(java.awt.event.ActionEvent evt
    ) {//GEN-FIRST:event_jButton3ActionPerformed
527     ArrayList<Produccion> prod = this.gramatica.getPr();
528     if(this.jTableSim.getRowCount()<2){
529         this.jButton3.setEnabled(false);
530     }else{
531         this.jButton5.setEnabled(true);
532         this.jButton3.setEnabled(true);
533         this.jButton2.setEnabled(true);
534         this.jButton4.setEnabled(true);
535         String pila = this.tabla.getValueAt(this.jTableSim.
            getRowCount()-2, 0).toString();
536         String entrada = this.tabla.getValueAt(this.jTableSim.
            getRowCount()-2, 1).toString();
537         this.tabla.removeRow(this.jTableSim.getRowCount()-1);
538
539         String[] p = pila.split(" ");
540         String[] e = entrada.split(" ");
541         int i = 1;
542         this.pila.removeAllElements();
543         while(i < p.length){
544             this.pila.push(p[i]);
545             i++;
546         }
547         i=e.length-1;
548         this.entrada.removeAllElements();
549         while(i > 0){
550             this.entrada.push(e[i]);
551             i--;
552         }
553
554         this.accion = this.buscarTablaAccion(Integer.parseInt(this.
            pila.peek()), this.entrada.peek());
555
556         if(this.accion.startsWith("r")){
557             Produccion pr = prod.get(Integer.parseInt(this.accion.
                substring(1, this.accion.length())));
558             this.accion = this.accion + " " + pr.getAntec().
                getSimboloNT().getNombre();
559             this.accion = this.accion + " \u2192 ";
560             i=0;
561             while(i < pr.getConsec().size()){
562                 this.accion = this.accion + " " + pr.getConsec().get(i
                    ).getNombre();
563                 i++;
564             }
565         }
566     }
567 }//GEN-LAST:event_jButton3ActionPerformed
568
569 private void jButton4ActionPerformed(java.awt.event.ActionEvent evt
    ) {//GEN-FIRST:event_jButton4ActionPerformed
570     this.siguientePaso();

```

```

571 } //GEN-LAST: event_jButton4ActionPerformed
572
573 private void jButton6ActionPerformed(java.awt.event.ActionEvent evt
574 ) { //GEN-FIRST: event_jButton6ActionPerformed
575     int conf = JOptionPane.showConfirmDialog(null, "¿Desea salir
576     ?" , "Salir",JOptionPane.YES_NO_OPTION);
577
578     if (conf==0)
579         this.dispose();
580 } //GEN-LAST: event_jButton6ActionPerformed
581
582 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt
583 ) { //GEN-FIRST: event_jButton1ActionPerformed
584     CadEntrada ent = new CadEntrada(this);
585     ent.setVisible(true);
586     ent.setLocationRelativeTo(null);
587 } //GEN-LAST: event_jButton1ActionPerformed
588
589 private void jButton5ActionPerformed(java.awt.event.ActionEvent evt
590 ) { //GEN-FIRST: event_jButton5ActionPerformed
591     int i=0;
592     while(i < 500){
593         if(!this.accion.equals("Aceptar") || !this.accion.equals("
594         Error") || !this.accion.contains("conf")){
595             this.siguientePaso();
596             i++;
597         }else
598             break;
599     }
600 } //GEN-LAST: event_jButton5ActionPerformed
601
602 private void jButton7ActionPerformed(java.awt.event.ActionEvent evt
603 ) { //GEN-FIRST: event_jButton7ActionPerformed
604     Boolean resultado= null;
605     FileNameExtensionFilter filtro= null;
606     JFileChooser selector= null;
607
608     if(this.jTextField1.getText().isEmpty()){
609         JOptionPane.showConfirmDialog(null, "La cadena de entrada
610         no puede estar vac a", "Error",JOptionPane.
611         CLOSED_OPTION);
612     }else{
613         JFileChooser chooser = new JFileChooser();
614         selector=chooser;
615         FileNameExtensionFilter extension = new
616             FileNameExtensionFilter(" Informes de simulacion
617             Ascendente (.pdf)",new String[]
618             {"pdf"});
619         filtro=extension;
620         selector.setFileFilter(filtro);
621         File fichero = new File("informeSimulacionAsc.pdf");
622         selector.setSelectedFile(fichero);
623         if (selector.showSaveDialog(null)==0) {

```

```

614         try {
615             resultado = this.ventanaPadre.generarInforme(
616                 selector.getSelectedFile().toString());
617         } catch (DocumentException ex) {
618             Logger.getLogger(Editor.class.getName()).log(Level.
619                 SEVERE, null, ex);
620         }
621         if(resultado.booleanValue()) {
622             StringBuilder JdecGenerated80 = new StringBuilder()
623                 ;
624         } else {
625             JOptionPane.showConfirmDialog(null,"El informe de
626                 la gramática no se puede generar hasta que la
627                 gramática esté validada.", "Informe de la
628                 gramática", JOptionPane.DEFAULT_OPTION);
629         }
630     }
631 }
632 }
633 }
634 }
635 }
636 }
637 }
638 }
639 }
640 }
641 }
642 }
643 }
644 }
645 }

```

*//GEN-LAST:event\_jButton7ActionPerformed*

*// Variables declaration – do not modify//GEN-BEGIN:variables*

```

631 private javax.swing.JButton jButton1;
632 private javax.swing.JButton jButton2;
633 private javax.swing.JButton jButton3;
634 private javax.swing.JButton jButton4;
635 private javax.swing.JButton jButton5;
636 private javax.swing.JButton jButton6;
637 private javax.swing.JButton jButton7;
638 private javax.swing.JLabel jLabel1;
639 private javax.swing.JLabel jLabel2;
640 private javax.swing.JPanel jPanel1;
641 private javax.swing.JScrollPane jScrollPane1;
642 private javax.swing.JTable jTableSim;
643 private javax.swing.JTextField jTextField1;
644 }
645 }

```

*// End of variables declaration//GEN-END:variables*

### 2.5.5. NuevaSimulacionDesc.java

```

1 //SimAS / Simulador
2 //Nueva Simulacion Descendente
3
4 package es.uco.simas.simulador;
5
6 import com.itextpdf.text.DocumentException;
7 import es.uco.simas.editor.Editor;
8 import es.uco.simas.editor.FuncionError;
9 import es.uco.simas.editor.TablaPredictiva;

```



```

10 import es.uco.simas.util.gramatica.Gramatica;
11 import es.uco.simas.util.gramatica.Simbolo;
12 import es.uco.simas.util.gramatica.Terminal;
13 import java.io.File;
14 import java.util.ArrayList;
15 import java.util.Stack;
16 import java.util.logging.Level;
17 import java.util.logging.Logger;
18 import javax.swing.DefaultListModel;
19 import javax.swing.JFileChooser;
20 import javax.swing.JOptionPane;
21 import javax.swing.filechooser.FileNameExtensionFilter;
22 import javax.swing.table.DefaultTableModel;
23
24 /**
25  * @author Vanesa
26  */
27 public class NuevaSimulacionDesc extends javax.swing.JFrame {
28
29     public Gramatica gramatica;
30     public Simulador ventanaPadre;
31     private DefaultTableModel tabla = new DefaultTableModel();
32     Stack<String> pila = new Stack<String>();
33     Stack<String> entrada = new Stack<String>();
34     private String accion = "";
35     private ArrayList<Simbolo> consec;
36
37     public NuevaSimulacionDesc(Gramatica gramatica, Simulador
38         ventanaPadre) {
39         initComponents();
40
41         this.gramatica = gramatica;
42         this.ventanaPadre = ventanaPadre;
43         this.jButton6.setEnabled(false);
44         if(this.ventanaPadre.getCadenaEntrada() != null){
45             iniciarSimulacion();
46         }
47         this.jTableSim.setDefaultRenderer (Object.class, new MiRender()
48             );
49         this.jTableSim.setModel(this.tabla);
50
51         this.jButton2.setEnabled(false);
52         this.jButton4.setEnabled(false);
53         this.jButton5.setEnabled(false);
54         this.jButton3.setEnabled(false);
55     }
56
57     /**
58      * This method is called from within the constructor to initialize
59      * the form.
60      * WARNING: Do NOT modify this code. The content of this method is
61      * always
62      * regenerated by the Form Editor.

```

```

59  */
60  @SuppressWarnings("unchecked")
61  // <editor-fold defaultstate="collapsed" desc="Generated Code">//
    GEN-BEGIN: initComponents
62  private void initComponents() {
63
64      jPanel1 = new javax.swing.JPanel();
65      jButton1 = new javax.swing.JButton();
66      jLabel1 = new javax.swing.JLabel();
67      jScrollPane1 = new javax.swing.JScrollPane();
68      jTableSim = new javax.swing.JTable();
69      jButton2 = new javax.swing.JButton();
70      jButton3 = new javax.swing.JButton();
71      jButton4 = new javax.swing.JButton();
72      jButton5 = new javax.swing.JButton();
73      jButton6 = new javax.swing.JButton();
74      jLabel2 = new javax.swing.JLabel();
75      jTextField1 = new javax.swing.JTextField();
76      jButton7 = new javax.swing.JButton();
77
78      setDefaultCloseOperation(javax.swing.WindowConstants.
        DISPOSE_ON_CLOSE);
79      setTitle("Simulacion Descendente");
80
81      jPanel1.setBackground(new java.awt.Color(233, 242, 242));
82
83      jButton1.setText("Cancelar");
84      jButton1.addActionListener(new java.awt.event.ActionListener()
85      {
86          public void actionPerformed(java.awt.event.ActionEvent evt)
87          {
88              jButton1ActionPerformed(evt);
89          }
90      });
91      jLabel1.setFont(new java.awt.Font("Tahoma", 1, 15)); // NOI18N
92      jLabel1.setText("Simulacion Descendente");
93
94      jTableSim.setModel(new javax.swing.table.DefaultTableModel(
95          new Object [][] {
96              {null, null, null},
97              {null, null, null},
98              {null, null, null},
99              {null, null, null}
100          },
101          new String [] {
102              "", "", ""
103          }
104      ) {
105          boolean[] canEdit = new boolean [] {
106              false, false, false
107          };

```

```

108         public boolean isCellEditable(int rowIndex, int columnIndex
109         ) {
110             return canEdit [columnIndex];
111         }
112     });
113     jScrollPane1.setViewportViewView(jTableSim);
114     jButton2.setIcon(new javax.swing.ImageIcon(getClass().
115         getResource("/es/uco/simas/resources/siguiente.png"))); //
116         NOI18N
117     jButton2.addActionListener(new java.awt.event.ActionListener()
118     {
119         public void actionPerformed(java.awt.event.ActionEvent evt)
120         {
121             jButton2ActionPerformed(evt);
122         }
123     });
124     jButton3.setIcon(new javax.swing.ImageIcon(getClass().
125         getResource("/es/uco/simas/resources/anterior.png"))); //
126         NOI18N
127     jButton3.addActionListener(new java.awt.event.ActionListener()
128     {
129         public void actionPerformed(java.awt.event.ActionEvent evt)
130         {
131             jButton3ActionPerformed(evt);
132         }
133     });
134     jButton4.setIcon(new javax.swing.ImageIcon(getClass().
135         getResource("/es/uco/simas/resources/ultimo.png"))); //
136         NOI18N
137     jButton4.addActionListener(new java.awt.event.ActionListener()
138     {
139         public void actionPerformed(java.awt.event.ActionEvent evt)
140         {
141             jButton4ActionPerformed(evt);
142         }
143     });
144     jButton5.setIcon(new javax.swing.ImageIcon(getClass().
145         getResource("/es/uco/simas/resources/primer.png"))); //
146         NOI18N
147     jButton5.addActionListener(new java.awt.event.ActionListener()
148     {
149         public void actionPerformed(java.awt.event.ActionEvent evt)
150         {
151             jButton5ActionPerformed(evt);
152         }
153     });
154     jButton6.setText("Informe de la Simulacion");

```

```

143     jButton6.addActionListener(new java.awt.event.ActionListener()
144     {
145         public void actionPerformed(java.awt.event.ActionEvent evt)
146         {
147             jButton6ActionPerformed(evt);
148         }
149     });
150
151     jLabel2.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
152     jLabel2.setText("Cadena de Entrada:");
153
154     jTextField1.setEditable(false);
155
156     jButton7.setText("Editar");
157     jButton7.addActionListener(new java.awt.event.ActionListener()
158     {
159         public void actionPerformed(java.awt.event.ActionEvent evt)
160         {
161             jButton7ActionPerformed(evt);
162         }
163     });
164
165     javax.swing.GroupLayout jPanel1Layout = new javax.swing.
166     GroupLayout(jPanel1);
167     jPanel1.setLayout(jPanel1Layout);
168     jPanel1Layout.setHorizontalGroup(
169         jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
170         Alignment.LEADING)
171         .addGroup(jPanel1Layout.createSequentialGroup()
172         .addGap(24, 24, 24)
173         .addComponent(jButton1)
174         .addPreferredGap(javax.swing.LayoutStyle.
175         ComponentPlacement.RELATED, javax.swing.GroupLayout.
176         DEFAULT_SIZE, Short.MAX_VALUE)
177         .addComponent(jButton6)
178         .addGap(41, 41, 41)
179         .addGroup(jPanel1Layout.createSequentialGroup()
180         .addGap(54, 54, 54)
181         .addGroup(jPanel1Layout.createParallelGroup(javax.swing
182         GroupLayout.Alignment.LEADING, false)
183         .addGroup(jPanel1Layout.createSequentialGroup()
184         .addGroup(jPanel1Layout.createParallelGroup(javax.swing.
185         GroupLayout.Alignment.TRAILING)
186         .addGroup(jPanel1Layout.
187         createSequentialGroup()
188         .addPreferredGap(javax.swing.
189         GroupLayout.Alignment.RELATED, javax.swing.GroupLayout.
190         DEFAULT_SIZE, Short.MAX_VALUE)
191         .addComponent(jButton5)
192         .addGap(18, 18, 18)
193         .addComponent(jButton3)
194         .addGap(18, 18, 18)

```

```

182         .addComponent(jButton2)
183         .addGap(18, 18, 18)
184         .addComponent(jButton4))
185     .addGroup(jPanel1Layout .
        createSequentialGroup ()
186         .addComponent(jLabel2)
187         .addPreferredGap(javax.swing .
            LayoutStyle .ComponentPlacement .
                UNRELATED)
            .addComponent(jTextField1)))
188     .addGap(18, 18, 18)
189     .addComponent(jButton7))
190     .addComponent(jScrollPane1 , javax.swing .GroupLayout .
        PREFERRED.SIZE, 484, javax.swing .GroupLayout .
        PREFERRED.SIZE))
191     .addContainerGap(37, Short.MAX_VALUE))
192 .addGroup(javax.swing .GroupLayout .Alignment .TRAILING,
193     jPanel1Layout .createSequentialGroup ()
194     .addContainerGap(javax.swing .GroupLayout .DEFAULT.SIZE,
        Short.MAX_VALUE)
195     .addComponent(jLabel1)
196     .addGap(165, 165, 165))
197 );
198 jPanel1Layout .setVerticalGroup(
199     jPanel1Layout .createParallelGroup(javax.swing .GroupLayout .
        Alignment .LEADING)
200     .addGroup(javax.swing .GroupLayout .Alignment .TRAILING,
        jPanel1Layout .createSequentialGroup ()
201         .addContainerGap ()
202         .addComponent(jLabel1)
203         .addGap(29, 29, 29)
204         .addGroup(jPanel1Layout .createParallelGroup(javax.swing
            .GroupLayout .Alignment .BASELINE)
            .addComponent(jLabel2)
205             .addComponent(jTextField1 , javax.swing .GroupLayout .
                PREFERRED.SIZE, javax.swing .GroupLayout .
                DEFAULT.SIZE, javax.swing .GroupLayout .
                PREFERRED.SIZE)
            .addComponent(jButton7))
206         .addPreferredGap(javax.swing .LayoutStyle .
            ComponentPlacement .RELATED, 44, Short.MAX_VALUE)
207         .addGroup(jPanel1Layout .createParallelGroup(javax.swing
            .GroupLayout .Alignment .LEADING)
            .addComponent(jButton5 , javax.swing .GroupLayout .
                Alignment .TRAILING)
208             .addComponent(jButton3 , javax.swing .GroupLayout .
                Alignment .TRAILING)
209             .addComponent(jButton2 , javax.swing .GroupLayout .
                Alignment .TRAILING)
210             .addComponent(jButton4 , javax.swing .GroupLayout .
                Alignment .TRAILING))
211         .addGap(31, 31, 31)

```

```

215         .addComponent(jScrollPane1 , javax.swing.GroupLayout.
                PREFERRED_SIZE, 319, javax.swing.GroupLayout.
                PREFERRED_SIZE)
216         .addPreferredGap(javax.swing.LayoutStyle.
                ComponentPlacement.UNRELATED)
217         .addGroup(jPanel1Layout.createParallelGroup(javax.swing
                .GroupLayout.Alignment.BASELINE)
218                 .addComponent(jButton1)
219                 .addComponent(jButton6))
220         .addGap(19, 19, 19))
221     );
222
223     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
        getContentPane());
224     getContentPane().setLayout(layout);
225     layout.setHorizontalGroup(
226         layout.createParallelGroup(javax.swing.GroupLayout.
            Alignment.LEADING)
227         .addComponent(jPanel1 , javax.swing.GroupLayout.
            PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE)
228     );
229     layout.setVerticalGroup(
230         layout.createParallelGroup(javax.swing.GroupLayout.
            Alignment.LEADING)
231         .addComponent(jPanel1 , javax.swing.GroupLayout.DEFAULT_SIZE
            , javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE
            )
232     );
233
234     pack();
235 } // </editor-fold> //GEN-END: initComponents
236
237 public void actualizarVisualizacion() {
238     String cadena = "";
239     this.pila.removeAllElements();
240     this.entrada.removeAllElements();
241     this.tabla = new DefaultTableModel();
242     this.jTableSim.setModel(this.tabla);
243
244     ArrayList<Terminal> cadenaEntrada = this.ventanaPadre.
        getCadenaEntrada();
245     int i = 0;
246     while (i < cadenaEntrada.size()) {
247         cadena = cadena+cadenaEntrada.get(i).getNombre()+" ";
248         i++;
249     }
250     this.jTextField1.setText(cadena);
251     if (!cadena.equals("")) {
252         this.iniciarSimulacion();
253     }
254     this.jButton2.setEnabled(true);
255     this.jButton4.setEnabled(true);

```

```

256         this.jButton5.setEnabled(false);
257         this.jButton3.setEnabled(false);
258     }
259
260     private void iniciarSimulacion() { //Con piLA
261         this.tabla.addColumn("Pila");
262         this.tabla.addColumn("Entrada");
263         this.tabla.addColumn("Accion");
264
265         ArrayList<Terminal> cadEntrada = this.ventanaPadre.
            getCadenaEntrada();
266         this.pila.push("$");
267         this.pila.push(this.gramatica.getSimbInicial());
268
269         int i=0;
270
271         this.entrada.push("$");
272         i = cadEntrada.size() - 1;
273         while(i >= 0){
274             this.entrada.push(cadEntrada.get(i).getNombre());
275             i--;
276         }
277
278         String accion = buscarTabla(this.pila.peek().toString(), this.
            entrada.peek().toString());
279         if(accion.equals("Error")){
280             this.jButton2.setEnabled(false);
281             this.jButton4.setEnabled(false);
282             this.jButton5.setEnabled(true);
283             this.jButton3.setEnabled(true);
284         }
285
286         Object [] linea = new Object[] {
287             mostrarPila(this.pila),
288             mostrarEntrada(this.entrada),
289             accion,
290         };
291
292         this.tabla.addRow(linea);
293         this.accion = accion;
294         this.jButton5.setEnabled(false);
295         this.jButton3.setEnabled(false);
296         this.jButton2.setEnabled(true);
297         this.jButton5.setEnabled(true);
298         this.jButton6.setEnabled(true);
299
300     }
301
302     private String buscarTabla(String fila, String columna){
303         DefaultTableModel tabla = this.gramatica.getTPredictiva().
            getTabla();
304         TablaPredictiva tpredictiva = this.gramatica.getTPredictiva();
305         int i=0;

```

```

306     int fil = -1;
307     int col = -1;
308     Object celda = "";
309
310     while(i < tabla.getRowCount()){
311         if(fila.equals(tpredictiva.getCeldaPredictiva(i, 0))){
312             fil = i;
313             break;
314         }else{
315             i++;
316         }
317     }
318     i = 1;
319     while(i < tabla.getColumnCount()){
320         if(columna.equals(tabla洗getColumnName(i))){
321             col = i;
322             break;
323         }else{
324             i++;
325         }
326     }
327     if(fil != -1 && col != -1){
328         celda = tpredictiva.getCeldaPredictiva(fil, col);
329     }
330     if(celda == null){
331         return "Error";
332     }else{
333         if(celda.toString().startsWith("E") && !celda.toString().
334             equals("Emparejar")){
335             return celda.toString();
336         }else{
337             String prod = buscarProduccion(Integer.parseInt(celda.
338                 toString()));
339             return (celda+" "+prod);
340         }
341     }
342 }
343
344 private String mostrarPila(Stack pila){
345     int i = 0;
346     String str = "";
347     while(i < pila.size()){
348         str = str + " "+pila.get(i);
349         i++;
350     }
351     return str;
352 }
353
354 private String mostrarEntrada(Stack pila){
355     int i = pila.size()-1;
356     String str = "";
357     while(i >= 0){
358         str = str + " "+pila.get(i);

```



```

357         i--;
358     }
359     return str;
360 }
361
362 private String buscarProduccion(int numero){
363     DefaultListModel prod = this.gramatica.getProducciones();
364     this.consec = this.gramatica.getPr().get(numero-1).getConsec();
365
366     return prod.elementAt(numero-1).toString();
367 }
368
369 private void siguientePaso(){
370     if(this.accion.equals("Aceptar") || this.accion.equals("Error")
371         || this.accion.equals("Fin")){
372         this.jButton2.setEnabled(false);
373         this.jButton4.setEnabled(false);
374         this.jButton3.setEnabled(true);
375         this.jButton5.setEnabled(true);
376     }else{
377         if(this.accion.equals("Emparejar")){
378             this.pila.pop();
379             this.entrada.pop();
380         }else{
381             if(this.accion.startsWith("E")){
382                 funError();
383             }else{
384                 if(this.consec.get(0).getNombre().equals("\u03b5")){
385                     {
386                         this.pila.pop();
387                     }else{
388                         int i=this.consec.size()-1;
389                         this.pila.pop();
390                         while(i >= 0){
391                             this.pila.push(consec.get(i).getNombre());
392                             i--;
393                         }
394                     }
395                 }
396             if(this.pila.peek().equals("$") && this.entrada.peek().
397                 equals("$")){
398                 this.accion = "Aceptar";
399             }else{
400                 if(this.pila.peek().equals(this.entrada.peek())){
401                     this.accion = "Emparejar";
402                 }else{
403                     this.accion = buscarTabla(this.pila.peek().toString
404                         (),this.entrada.peek().toString());
405                 }
406             }
407         }
408     }
409 }

```

```

406         Object [] linea = new Object [] {
407             mostrarPila(this.pila),
408             mostrarEntrada(this.entrada),
409             this.accion,
410         };
411
412         this.tabla.addRow(linea);
413     }
414 }
415
416 private void funError() {
417     ArrayList<FuncionError> funError = this.gramatica.
418         getTPredictiva().getFunError();
419     FuncionError fun = new FuncionError();
420     int i = 0;
421     int num = Integer.parseInt(this.accion.substring(1));
422
423     while(i < funError.size()){
424         if(funError.get(i).getIdentificador() == num){
425             fun = funError.get(i);
426             break;
427         } else
428             i++;
429     }
430
431     if(fun.getAccion() == 1){
432         this.entrada.push(fun.getSimbolo().getNombre());
433     }
434     if(fun.getAccion() == 2){
435         this.entrada.pop();
436     }
437     if(fun.getAccion() == 3){
438         this.entrada.pop();
439         this.entrada.push(fun.getSimbolo().getNombre());
440     }
441     if(fun.getAccion() == 4){
442         this.pila.push(fun.getSimbolo().getNombre());
443     }
444     if(fun.getAccion() == 5){
445         this.pila.pop();
446     }
447     if(fun.getAccion() == 6){
448         this.pila.pop();
449         this.pila.push(fun.getSimbolo().getNombre());
450     }
451     if(fun.getAccion() == 7){
452         this.accion = "Fin";
453     }
454 }
455
456 public DefaultTableModel getTabla() {
457     this jButton4ActionPerformed(null);
458     return this.tabla;
459 }

```

```

458     }
459
460     private void jButton1ActionPerformed(java.awt.event.ActionEvent evt
461     ) {//GEN-FIRST:event_jButton1ActionPerformed
462         int conf = JOptionPane.showConfirmDialog(null, "¿Desea salir
463         ?", "Salir",JOptionPane.YES_NO_OPTION);
464
465         if (conf==0)
466             this.dispose();
467     }//GEN-LAST:event_jButton1ActionPerformed
468
469     private void jButton2ActionPerformed(java.awt.event.ActionEvent evt
470     ) {//GEN-FIRST:event_jButton2ActionPerformed
471         siguientePaso();
472         this.jButton5.setEnabled(true);
473         this.jButton3.setEnabled(true);
474         this.jButton2.setEnabled(true);
475         this.jButton4.setEnabled(true);
476     }//GEN-LAST:event_jButton2ActionPerformed
477
478     private void jButton5ActionPerformed(java.awt.event.ActionEvent evt
479     ) {//GEN-FIRST:event_jButton5ActionPerformed
480         DefaultTableModel tabla2 = new DefaultTableModel();
481         this.tabla = tabla2;
482         this.jTableSim.setModel(this.tabla);
483         this.pila.removeAllElements();
484         this.entrada.removeAllElements();
485         this.accion = "";
486         this.iniciarSimulacion();
487         this.jButton5.setEnabled(false);
488         this.jButton3.setEnabled(false);
489         this.jButton2.setEnabled(true);
490         this.jButton4.setEnabled(true);
491     }//GEN-LAST:event_jButton5ActionPerformed
492
493     private void jButton3ActionPerformed(java.awt.event.ActionEvent evt
494     ) {//GEN-FIRST:event_jButton3ActionPerformed
495         if(this.jTableSim.getRowCount()<2){
496             this.jButton3.setEnabled(false);
497         }else{
498             this.jButton5.setEnabled(true);
499             this.jButton3.setEnabled(true);
500             this.jButton2.setEnabled(true);
501             this.jButton4.setEnabled(true);
502             String pila = this.tabla.getValueAt(this.jTableSim.
503             getRowCount()-2, 0).toString();
504             String entrada = this.tabla.getValueAt(this.jTableSim.
505             getRowCount()-2, 1).toString();
506             this.tabla.removeRow(this.jTableSim.getRowCount()-1);
507
508             String [] p = pila.split(" ");
509             String [] e = entrada.split(" ");
510             int i = 1;

```

```

504         this.pila.removeAllElements();
505         while(i < p.length ){
506             this.pila.push(p[i]);
507             i++;
508         }
509         i=e.length-1;
510         this.entrada.removeAllElements();
511         while(i > 0 ){
512             this.entrada.push(e[i]);
513             i--;
514         }
515
516         if( this.pila.peek().equals("$") && this.entrada.peek().
            equals("$")){
517             this.accion = "Aceptar";
518         } else {
519             if( this.pila.peek().equals(this.entrada.peek())){
520                 this.accion = "Emparejar";
521             } else {
522                 this.accion = buscarTabla(this.pila.peek().toString
                    (), this.entrada.peek().toString());
523             }
524         }
525     }
526 } //GEN-LAST: event_jButton3ActionPerformed
527
528 private void jButton4ActionPerformed(java.awt.event.ActionEvent evt
529 ) { //GEN-FIRST: event_jButton4ActionPerformed
530     int i=0;
531     while(i < 500){
532         if(!this.accion.equals("Aceptar") || !this.accion.equals("
533             Error") || !this.accion.equals("Fin")){
534             this.siguientePaso();
535             i++;
536         } else
537             break;
538     }
539     this.jButton3.setEnabled(true);
540     this.jButton5.setEnabled(true);
541     this.jButton2.setEnabled(false);
542     this.jButton4.setEnabled(false);
543 } //GEN-LAST: event_jButton4ActionPerformed
544
545 private void jButton6ActionPerformed(java.awt.event.ActionEvent evt
546 ) { //GEN-FIRST: event_jButton6ActionPerformed
547
548     Boolean resultado= null;
549     FileNameExtensionFilter filtro= null;
550     JFileChooser selector= null;
551
552     JFileChooser chooser = new JFileChooser();
553     selector=chooser;

```

```

551     FileNameExtensionFilter extension = new FileNameExtensionFilter
552         ("Informe de simulacion Descendente (.pdf)", new String[]
553             {"pdf"});
554     filtro=extension;
555     selector.setFileFilter(filtro);
556     File fichero = new File("informeSimulacionDesc.pdf");
557     selector.setSelectedFile(fichero);
558     if(selector.showSaveDialog(null)==0) {
559         try {
560             resultado = this.ventanaPadre.generarInforme(selector.
561                 getSelectedFile().toString());
562         } catch (DocumentException ex) {
563             Logger.getLogger(Editor.class.getName()).log(Level.
564                 SEVERE, null, ex);
565         }
566         if(resultado.booleanValue()) {
567             StringBuilder JdecGenerated80 = new StringBuilder();
568         } else {
569             JOptionPane.showConfirmDialog(null, "El informe de la
570                 gramática no se puede generar hasta que la
571                 gramática esté validada.", "Informe de la
572                 gramática", JOptionPane.DEFAULT_OPTION);
573         }
574     }
575 }
576 }
577 }
578 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
590 }
591 }
592 }
593 }
594 }

```

### 2.5.6. PanelNuevaSimAscPaso1.java

```

1 //SimAS / Simulador
2 //Panel Nueva Simulacion Ascendente paso 1
3
4 package es.uco.simas.simulador;
5
6 import javax.swing.JOptionPane;
7
8 /**
9  * @author vanesa
10  */
11 public class PanelNuevaSimAscPaso1 extends javax.swing.JPanel {
12
13     private VentanaSimuladorAsc ventanaPadre;
14     int metodo = -1;
15
16     public PanelNuevaSimAscPaso1(VentanaSimuladorAsc ventanaPadre) {
17         initComponents();
18         this.ventanaPadre = ventanaPadre;
19     }
20
21     /**
22      * This method is called from within the constructor to initialize
23      * the form.
24      * WARNING: Do NOT modify this code. The content of this method is
25      * always
26      * regenerated by the Form Editor.
27      */
28     // <editor-fold defaultstate="collapsed" desc="Generated Code">
29     private void initComponents() {
30
31         jLabel1 = new javax.swing.JLabel();
32         jButtonCancelar = new javax.swing.JButton();
33         jButton2 = new javax.swing.JButton();
34         jButton3 = new javax.swing.JButton();
35         jButton4 = new javax.swing.JButton();
36         jButton5 = new javax.swing.JButton();
37         jComboBox1 = new javax.swing.JComboBox();
38
39         setBackground(new java.awt.Color(233, 244, 244));
40         setToolTipText("Simulacion Ascendente. Paso 1.");
41
42         jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
43         jLabel1.setText("Elige el MÃtodo de simulacion Ascendente:");
44
45         jButtonCancelar.setText("Cancelar");
46         jButtonCancelar.addActionListener(new java.awt.event.
47             ActionListener() {

```

```

47         public void actionPerformed(java.awt.event.ActionEvent evt)
48         {
49             jButtonCancelarActionPerformed(evt);
50         }
51     });
52     jButton2.setIcon(new javax.swing.ImageIcon(getClass().
53         getResource("/es/uco/simas/resources/ultimo.png"))); //
54         NOI18N
55     jButton2.addActionListener(new java.awt.event.ActionListener()
56     {
57         public void actionPerformed(java.awt.event.ActionEvent evt)
58         {
59             jButton2ActionPerformed(evt);
60         }
61     });
62     jButton3.setIcon(new javax.swing.ImageIcon(getClass().
63         getResource("/es/uco/simas/resources/siguiente.png"))); //
64         NOI18N
65     jButton3.addActionListener(new java.awt.event.ActionListener()
66     {
67         public void actionPerformed(java.awt.event.ActionEvent evt)
68         {
69             jButton3ActionPerformed(evt);
70         }
71     });
72     jButton4.setIcon(new javax.swing.ImageIcon(getClass().
73         getResource("/es/uco/simas/resources/anterior.png"))); //
74         NOI18N
75     jButton4.setEnabled(false);
76     jButton5.setIcon(new javax.swing.ImageIcon(getClass().
77         getResource("/es/uco/simas/resources/primer.png"))); //
78         NOI18N
79     jButton5.setEnabled(false);
80     JComboBox1.setModel(new javax.swing.DefaultComboBoxModel(new
81         String[] { "MÃ©todo SLR", "MÃ©todo LR-Canónico", "MÃ©todo
82         LALR" }));
83     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
84         this);
85     this.setLayout(layout);
86     layout.setHorizontalGroup(
87         layout.createParallelGroup(javax.swing.GroupLayout.
88             Alignment.LEADING)
89         .addGroup(layout.createSequentialGroup()
90             .addGap(152, 152, 152)
91             .addComponent(jButton1, javax.swing.GroupLayout.
92                 PREFERRED_SIZE, javax.swing.GroupLayout.
93                 DEFAULT_SIZE, true)
94             .addComponent(jButton2, javax.swing.GroupLayout.
95                 PREFERRED_SIZE, javax.swing.GroupLayout.
96                 DEFAULT_SIZE, true)
97             .addComponent(jButton3, javax.swing.GroupLayout.
98                 PREFERRED_SIZE, javax.swing.GroupLayout.
99                 DEFAULT_SIZE, true)
100             .addComponent(jButton4, javax.swing.GroupLayout.
101                 PREFERRED_SIZE, javax.swing.GroupLayout.
102                 DEFAULT_SIZE, true)
103             .addComponent(jButton5, javax.swing.GroupLayout.
104                 PREFERRED_SIZE, javax.swing.GroupLayout.
105                 DEFAULT_SIZE, true)
106             .addComponent(JComboBox1, javax.swing.GroupLayout.
107                 PREFERRED_SIZE, javax.swing.GroupLayout.
108                 DEFAULT_SIZE, true)
109             .addContainerGap(152, true)
110         )
111     );
112     javax.swing.GroupLayout.setDefaultLookAndFeelDecorations(true);
113     this.setDefaultCloseOperation(javax.swing.WindowConstants.
114         DISPOSE_ON_CLOSE);
115     setTitle("Simulador de compiladores");
116     setSize(800, 600);
117     setLocationRelativeTo(null);
118     setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
119     getContentPane().setLayout(layout);
120     layout.linkSize(javax.swing.SwingConstants.HORIZONTAL,
121         new java.awt.Component[] { jButton1, jButton2, jButton3,
122             jButton4, jButton5, JComboBox1 });
123     layout.setVerticalGroup(
124         layout.createParallelGroup(javax.swing.GroupLayout.
125             Alignment.LEADING)
126         .addGroup(layout.createSequentialGroup()
127             .add(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE,
128                 25, javax.swing.GroupLayout.PREFERRED_SIZE)
129             .add(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE,
130                 25, javax.swing.GroupLayout.PREFERRED_SIZE)
131             .add(jButton3, javax.swing.GroupLayout.PREFERRED_SIZE,
132                 25, javax.swing.GroupLayout.PREFERRED_SIZE)
133             .add(jButton4, javax.swing.GroupLayout.PREFERRED_SIZE,
134                 25, javax.swing.GroupLayout.PREFERRED_SIZE)
135             .add(jButton5, javax.swing.GroupLayout.PREFERRED_SIZE,
136                 25, javax.swing.GroupLayout.PREFERRED_SIZE)
137             .add(JComboBox1, javax.swing.GroupLayout.PREFERRED_SIZE,
138                 25, javax.swing.GroupLayout.PREFERRED_SIZE)
139             .addContainerGap(152, true)
140         )
141     );
142     pack();
143 
```

```

81         PREFERRED.SIZE)
82         .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
83         Short.MAX_VALUE))
84     .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
85     layout.createSequentialGroup())
86     .addGroup(layout.createParallelGroup(javax.swing.
87     GroupLayout.Alignment.TRAILING)
88     .addGroup(layout.createSequentialGroup())
89     .addContainerGap(29, Short.MAX_VALUE)
90     .addComponent(jLabel1))
91     .addGroup(layout.createSequentialGroup())
92     .addGap(41, 41, 41)
93     .addComponent(jButtonCancelar)
94     .addPreferredGap(javax.swing.LayoutStyle.
95     ComponentPlacement.RELATED, javax.swing.
96     GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
97     .addComponent(jButton5)
98     .addPreferredGap(javax.swing.LayoutStyle.
99     ComponentPlacement.RELATED)
100    .addComponent(jButton4)
101    .addPreferredGap(javax.swing.LayoutStyle.
102    ComponentPlacement.RELATED)
103    .addComponent(jButton3)))
104    .addPreferredGap(javax.swing.LayoutStyle.
105    ComponentPlacement.RELATED)
106    .addComponent(jButton2)
107    .addGap(18, 18, 18))
108    );
109    layout.setVerticalGroup(
110    layout.createParallelGroup(javax.swing.GroupLayout.
111    Alignment.LEADING)
112    .addGroup(layout.createSequentialGroup())
113    .addGap(67, 67, 67)
114    .addComponent(jLabel1)
115    .addGap(53, 53, 53)
116    .addComponent(jComboBox1, javax.swing.GroupLayout.
117    PREFERRED.SIZE, javax.swing.GroupLayout.
118    DEFAULT_SIZE, javax.swing.GroupLayout.
119    PREFERRED.SIZE)
120    .addPreferredGap(javax.swing.LayoutStyle.
121    ComponentPlacement.RELATED, 117, Short.MAX_VALUE)
122    .addGroup(layout.createParallelGroup(javax.swing.
123    GroupLayout.Alignment.LEADING)
124    .addComponent(jButton2)
125    .addGroup(layout.createParallelGroup(javax.swing.
126    GroupLayout.Alignment.BASELINE)
127    .addComponent(jButtonCancelar)
128    .addComponent(jButton3)
129    .addComponent(jButton4)
130    .addComponent(jButton5)))
131    .addGap(24, 24, 24))
132    );

```



```

118         getAccessibleContext().setAccessibleParent(jButtonCancelar);
119     }// </editor-fold>//GEN-END: initComponents
120
121     private void jButtonCancelarActionPerformed(java.awt.event.
        ActionEvent evt) {//GEN-FIRST:
        event_jButtonCancelarActionPerformed
122         int conf = JOptionPane.showConfirmDialog(null, "¿Desea salir
            del asistente de la simulacion de la gramática?", "Salir",
            JOptionPane.YES_NO_OPTION);
123
124         if (conf==0)
125             this.ventanaPadre.dispose();
126     }//GEN-LAST: event_jButtonCancelarActionPerformed
127
128     private void jButton3ActionPerformed(java.awt.event.ActionEvent evt
        ) {//GEN-FIRST: event_jButton3ActionPerformed
129         this.metodo = this.jComboBox1.getSelectedIndex();
130         this.ventanaPadre.setMetodo(this.metodo);
131         this.ventanaPadre.cambiarPaso(2);
132     }//GEN-LAST: event_jButton3ActionPerformed
133
134     private void jButton2ActionPerformed(java.awt.event.ActionEvent evt
        ) {//GEN-FIRST: event_jButton2ActionPerformed
135
136         this.metodo = this.jComboBox1.getSelectedIndex();
137         this.ventanaPadre.setMetodo(this.metodo);
138
139         this.ventanaPadre.gramatica.generarConjPrim();
140         this.ventanaPadre.gramatica.generarConjSig();
141         this.ventanaPadre.paso2.construirConjuntos(this.ventanaPadre.
            gramatica);
142
143         PanelNuevaSimAscPaso3 paso3 = new PanelNuevaSimAscPaso3(this.
            ventanaPadre);
144
145         PanelNuevaSimAscPaso4 paso4 = new PanelNuevaSimAscPaso4(this.
            ventanaPadre);
146
147         this.ventanaPadre.cambiarPaso(5);
148
149     }//GEN-LAST: event_jButton2ActionPerformed
150
151     // Variables declaration - do not modify//GEN-BEGIN: variables
152     private javax.swing.JButton jButton2;
153     private javax.swing.JButton jButton3;
154     private javax.swing.JButton jButton4;
155     private javax.swing.JButton jButton5;
156     private javax.swing.JButton jButtonCancelar;
157     private javax.swing.JComboBox jComboBox1;
158     private javax.swing.JLabel jLabel1;
159     // End of variables declaration//GEN-END: variables
160 }

```

### 2.5.7. PanelNuevaSimAscPaso2.java

```

1 //SimAS / Simulador
2 //Panel Nueva Simulacion Ascendente paso 2
3
4 package es.uco.simas.simulador;
5
6 import es.uco.simas.util.gramatica.Gramatica;
7 import es.uco.simas.util.gramatica.NoTerminal;
8 import es.uco.simas.util.gramatica.Terminal;
9 import java.util.ArrayList;
10 import java.util.Iterator;
11 import javax.swing.JOptionPane;
12 import javax.swing.JTable;
13 import javax.swing.table.DefaultTableModel;
14
15 /**
16  * @author vanesa
17  */
18 public class PanelNuevaSimAscPaso2 extends javax.swing.JPanel {
19
20     private VentanaSimuladorAsc ventanaPadre;
21     private DefaultTableModel modeloConjuntos ;
22     private JTable tblConjuntos ;
23
24     public PanelNuevaSimAscPaso2(VentanaSimuladorAsc ventanaPadre) {
25         DefaultTableModel tabla = new DefaultTableModel();
26         this.modeloConjuntos =tabla;
27         initComponents();
28
29         this.ventanaPadre =ventanaPadre;
30
31         this.jTable1.getColumnModel().getColumn(0).setPreferredWidth
32             (90);
33         this.jTable1.getColumnModel().getColumn(1).setPreferredWidth
34             (250);
35         this.jTable1.getColumnModel().getColumn(2).setPreferredWidth
36             (290);
37         this.jTable1.setEnabled(false);
38         Gramatica gr = this.ventanaPadre.gramatica;
39
40     }
41
42     public void construirConjuntos(Gramatica gramatica) {
43         ArrayList<com.itextpdf.text.Paragraph>
44             construccionConjuntoPrimero= new ArrayList();
45         ArrayList<com.itextpdf.text.Paragraph>
46             construccionConjuntoSiguiente= new ArrayList();
47         ArrayList<NoTerminal> noTerminales= null;
48         ArrayList<Terminal> conjuntoPrimero= null;
49         ArrayList<Terminal> conjuntoSiguiente= null;
50         ArrayList<Terminal> conjuntoPrimeroSimbolo= null;
51         ArrayList<Terminal> conjuntoSiguienteSimbolo= null;

```

```

47      Iterator i$= null;
48      NoTerminal noTerminal= null;
49      String strConjuntoPrimero= null;
50      String strConjuntoSiguiente= null;
51
52      int i= 0;
53      int j=0;
54
55      this.modeloConjuntos =new DefaultTableModel();
56      this.modeloConjuntos.addColumn(" Simbolo");
57      this.modeloConjuntos.addColumn(" Conjunto primero");
58      this.modeloConjuntos.addColumn(" Conjunto siguiente");
59      this.jTable1.setModel( this.modeloConjuntos);
60      this.jTable1.getColumnModel().getColumn(0).setPreferredWidth
        (95);
61      this.jTable1.getColumnModel().getColumn(1).setPreferredWidth
        (250);
62      this.jTable1.getColumnModel().getColumn(2).setPreferredWidth
        (290);
63      this.jTable1.setEnabled( false);
64
65      noTerminales = gramatica.getNoTerm();
66
67      i=0;
68      while(i < noTerminales.size()){
69          strConjuntoPrimero = "";
70          strConjuntoSiguiente = "";
71          noTerminal = noTerminales.get(i);
72          conjuntoPrimero = noTerminal.getPrimeros();
73          conjuntoSiguiente = noTerminal.getSiguientes();
74
75          j=0;
76          while(j < conjuntoPrimero.size()){
77              StringBuilder cadena = new StringBuilder();
78              strConjuntoPrimero = cadena.append(strConjuntoPrimero).
                append(" ").append(conjuntoPrimero.get(j).getNombre
                ()).toString();
79
80              j++;
81          }
82
83          j=0;
84          while(j < conjuntoSiguiente.size()){
85              StringBuilder cadena = new StringBuilder();
86              strConjuntoSiguiente = cadena.append(
                strConjuntoSiguiente).append(" ").append(
                conjuntoSiguiente.get(j).getNombre()).toString();
87
88              j++;
89          }
90
91          Object [] linea = new Object []
92          {

```

```

93         noTerminal.getValor(),          strConjuntoPrimero,
94         strConjuntoSiguiente
95     };
96     this.modeloConjuntos.addRow(linea);
97     i++;
98 }
99     this.ventanaPadre.gramatica = gramatica;
100 }
101 /**
102  * This method is called from within the constructor to initialize
103  * the form.
104  * WARNING: Do NOT modify this code. The content of this method is
105  * always
106  * regenerated by the Form Editor.
107  */
108 @SuppressWarnings("unchecked")
109 // <editor-fold defaultstate="collapsed" desc="Generated Code">
110 GEN-BEGIN: initComponents
111 private void initComponents() {
112     jScrollPane1 = new javax.swing.JScrollPane();
113     jTable1 = new javax.swing.JTable();
114     jLabel1 = new javax.swing.JLabel();
115     jButtonCancelar = new javax.swing.JButton();
116     jButtonPrimero = new javax.swing.JButton();
117     jButtonAnterior = new javax.swing.JButton();
118     jButtonSiguiente = new javax.swing.JButton();
119     jButtonUltimo = new javax.swing.JButton();
120
121     setBackground(new java.awt.Color(233, 244, 244));
122
123     jTable1.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
124     jTable1.setModel(new javax.swing.table.DefaultTableModel(
125         new Object [][] {
126             {null, null, null},
127             {null, null, null},
128             {null, null, null},
129             {null, null, null},
130             {null, null, null}
131         },
132         new String [] {
133             "S mbolo", "Conjunto Primero", "Conjunto Siguiente"
134         }
135     ) {
136         boolean[] canEdit = new boolean [] {
137             false, false, false
138         };
139
140         public boolean isCellEditable(int rowIndex, int columnIndex) {
141             return canEdit [columnIndex];
142         }
143     }

```

```

141     });
142     jScrollPane1.setViewportViewView(jTable1);
143
144     jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
145     jLabel1.setText("Conjunto Primero y Siguiente");
146
147     jButtonCancelar.setText("Cancelar");
148     jButtonCancelar.addActionListener(new java.awt.event.
        ActionListener() {
149         public void actionPerformed(java.awt.event.ActionEvent evt)
        {
150             jButtonCancelarActionPerformed(evt);
151         }
152     });
153
154     jButtonPrimero.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/primero.png"))); //
        NOI18N
155     jButtonPrimero.addActionListener(new java.awt.event.
        ActionListener() {
156         public void actionPerformed(java.awt.event.ActionEvent evt)
        {
157             jButtonPrimeroActionPerformed(evt);
158         }
159     });
160
161     jButtonAnterior.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/anterior.png"))); //
        NOI18N
162     jButtonAnterior.addActionListener(new java.awt.event.
        ActionListener() {
163         public void actionPerformed(java.awt.event.ActionEvent evt)
        {
164             jButtonAnteriorActionPerformed(evt);
165         }
166     });
167
168     jButtonSiguiente.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/siguiente.png"))); //
        NOI18N
169     jButtonSiguiente.addActionListener(new java.awt.event.
        ActionListener() {
170         public void actionPerformed(java.awt.event.ActionEvent evt)
        {
171             jButtonSiguienteActionPerformed(evt);
172         }
173     });
174
175     jButtonUltimo.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/ultimo.png"))); //
        NOI18N
176     jButtonUltimo.addActionListener(new java.awt.event.
        ActionListener() {

```

```

177         public void actionPerformed(java.awt.event.ActionEvent evt)
178         {
179             jButtonUltimoActionPerformed(evt);
180         }
181     });
182     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
183         this);
184     this.setLayout(layout);
185     layout.setHorizontalGroup(
186         layout.createParallelGroup(javax.swing.GroupLayout.
187             Alignment.LEADING)
188             .addGroup(layout.createSequentialGroup()
189                 .addGroup(layout.createParallelGroup(javax.swing.
190                     GroupLayout.Alignment.LEADING)
191                     .addGroup(layout.createSequentialGroup()
192                         .addGap(126, 126, 126)
193                         .addComponent(jLabel1))
194                     .addGroup(layout.createSequentialGroup()
195                         .addGap(21, 21, 21)
196                         .addGroup(layout.createParallelGroup(javax.
197                             swing.GroupLayout.Alignment.LEADING, false)
198                             .addComponent(jScrollPane1, javax.swing.
199                                 GroupLayout.PREFERRED_SIZE, 575, javax.
200                                     swing.GroupLayout.PREFERRED_SIZE)
201                             .addGroup(layout.createSequentialGroup()
202                                 .addComponent(jButtonCancelar)
203                                 .addPreferredGap(javax.swing.
204                                     LayoutStyle.ComponentPlacement.
205                                         RELATED, javax.swing.GroupLayout.
206                                             DEFAULT_SIZE, Short.MAX_VALUE)
207                                 .addComponent(jButtonPrimero)
208                                 .addPreferredGap(javax.swing.
209                                     LayoutStyle.ComponentPlacement.
210                                         RELATED)
211                                 .addComponent(jButtonAnterior)
212                                 .addPreferredGap(javax.swing.
213                                     LayoutStyle.ComponentPlacement.
214                                         RELATED)
215                                 .addComponent(jButtonSiguiente)
216                                 .addPreferredGap(javax.swing.
217                                     LayoutStyle.ComponentPlacement.
218                                         RELATED)
219                                 .addComponent(jButtonUltimo))))))
220                 .addContainerGap(36, Short.MAX_VALUE))
221     );
222     layout.setVerticalGroup(
223         layout.createParallelGroup(javax.swing.GroupLayout.
224             Alignment.LEADING)
225             .addGroup(layout.createSequentialGroup()
226                 .addGroup(layout.createSequentialGroup()
227                     .addGap(42, 42, 42)
228                     .addComponent(jLabel1)
229                     .addGap(36, 36, 36)

```

```

213         .addComponent(jScrollPane1 , javax.swing.GroupLayout.
                PREFERRED_SIZE, 274, javax.swing.GroupLayout.
                PREFERRED_SIZE)
214         .addPreferredGap(javax.swing.LayoutStyle.
                ComponentPlacement.RELATED, 26, Short.MAX_VALUE)
215         .addGroup(layout.createParallelGroup(javax.swing.
                GroupLayout.Alignment.LEADING)
216                 .addGroup(layout.createParallelGroup(javax.swing.
                GroupLayout.Alignment.TRAILING)
                .addComponent(jButtonAnterior)
                .addComponent(jButtonSiguiente)
                .addComponent(jButtonUltimo))
                .addComponent(jButtonCancelar)
                .addGroup(layout.createSequentialGroup()
                .addGroup(layout.createParallelGroup()
                .addGroup(layout.createSequentialGroup()
                .addGap(2, 2, 2)
                .addComponent(jButtonPrimero)))
                .addGap(18, 18, 18))
                .addGap(18, 18, 18))
225     );
226 }// </editor-fold> //GEN-END: initComponents
227
228 private void jButtonCancelarActionPerformed(java.awt.event.
    ActionEvent evt) { //GEN-FIRST:
    event_jButtonCancelarActionPerformed
229     int conf = JOptionPane.showConfirmDialog(null, "¿Desea salir
        del asistente de la simulacion de la gramática?", "Salir",
        JOptionPane.YES_NO_OPTION);
230     if (conf==0)
231         this.ventanaPadre.dispose();
232 }//GEN-LAST: event_jButtonCancelarActionPerformed
233
234 private void jButtonPrimeroActionPerformed(java.awt.event.
    ActionEvent evt) { //GEN-FIRST:
    event_jButtonPrimeroActionPerformed
235     this.ventanaPadre.cambiarPaso(1);
236 }//GEN-LAST: event_jButtonPrimeroActionPerformed
237
238 private void jButtonAnteriorActionPerformed(java.awt.event.
    ActionEvent evt) { //GEN-FIRST:
    event_jButtonAnteriorActionPerformed
239     this.ventanaPadre.cambiarPaso(1);
240 }//GEN-LAST: event_jButtonAnteriorActionPerformed
241
242 private void jButtonSiguienteActionPerformed(java.awt.event.
    ActionEvent evt) { //GEN-FIRST:
    event_jButtonSiguienteActionPerformed
243     this.ventanaPadre.cambiarPaso(3);
244 }//GEN-LAST: event_jButtonSiguienteActionPerformed
245
246 private void jButtonUltimoActionPerformed(java.awt.event.
    ActionEvent evt) { //GEN-FIRST:
    event_jButtonUltimoActionPerformed
247

```

```

248         PanelNuevaSimAscPaso3 paso3 = new PanelNuevaSimAscPaso3(this.
                ventanaPadre);
249
250         PanelNuevaSimAscPaso4 paso4 = new PanelNuevaSimAscPaso4(this.
                ventanaPadre);
251
252         this.ventanaPadre.cambiarPaso(5);
253     } //GEN-LAST: event_jButtononUltimoActionPerformed
254
255     // Variables declaration - do not modify //GEN-BEGIN: variables
256     private javax.swing.JButton jButtonAnterior;
257     private javax.swing.JButton jButtonCancelar;
258     private javax.swing.JButton jButtonPrimero;
259     private javax.swing.JButton jButtonSiguiente;
260     private javax.swing.JButton jButtonUltimo;
261     private javax.swing.JLabel jLabel1;
262     private javax.swing.JScrollPane jScrollPane1;
263     private javax.swing.JTable jTable1;
264     // End of variables declaration //GEN-END: variables
265 }

```

### 2.5.8. PanelNuevaSimAscPaso3.java

```

1  //SimAS / Simulador
2  //Panel Nueva Simulacion Ascendente paso 3
3
4  package es.uco.simas.simulador;
5
6  import es.uco.simas.editor.ColCanLALR;
7  import es.uco.simas.editor.ColCanLR0;
8  import es.uco.simas.editor.ColCanLR1;
9  import javax.swing.JOptionPane;
10
11 /**
12  * @author vanesa
13  */
14 public class PanelNuevaSimAscPaso3 extends javax.swing.JPanel {
15
16     private VentanaSimuladorAsc ventanaPadre;
17
18     public PanelNuevaSimAscPaso3(VentanaSimuladorAsc ventanaPadre) {
19         initComponents();
20         this.ventanaPadre = ventanaPadre;
21         if (this.ventanaPadre.getMetodo() == 0) {
22             this.jLabel1.setText(" Coleccion Canonica Elementos LR(0)");
23             if (this.ventanaPadre.getGramatica().getColeccionLR0() == null)
24                 ColCanLR0 col = new ColCanLR0(this.ventanaPadre.
25                     getGramatica());
26                 col.construir();

```



```

26         this.ventanaPadre.getGramatica().setColeccionLR0(col);
27         this.ventanaPadre.setColeccion(col);
28     }
29     this.jTextArea1.setText(this.ventanaPadre.getGramatica
        ().getColeccionLR0().getColeccion());
30 }
31 if(this.ventanaPadre.getMetodo() == 1){
32     this.jLabel1.setText("Coleccion Canonica LR(1)-elementos");
33     if(this.ventanaPadre.getGramatica().getColeccionLR1()==null)
34     {
35         ColCanLR1 col = new ColCanLR1(this.ventanaPadre.
36             getGramatica());
37         col.construir();
38         this.ventanaPadre.getGramatica().setColeccionLR1(col);
39         this.ventanaPadre.setColeccion(col);
40     }
41     this.jTextArea1.setText(this.ventanaPadre.getGramatica
42         ().getColeccionLR1().getColeccion());
43 }
44 if(this.ventanaPadre.getMetodo() == 2){
45     this.jLabel1.setText("Coleccion Canonica Elementos LALR(1)"
46         );
47     ColCanLR1 col = new ColCanLR1();
48     if(this.ventanaPadre.getGramatica().getColeccionLALR()==
49         null){
50         if(this.ventanaPadre.getGramatica().getColeccionLR1()
51             == null){
52             col = new ColCanLR1(this.ventanaPadre.getGramatica
53                 ());
54             col.construir();
55         }else{
56             col = this.ventanaPadre.getGramatica().
57                 getColeccionLR1();
58         }
59     }
60     ColCanLALR colLALR = new ColCanLALR(this.ventanaPadre.
61         getGramatica(), col);
62     colLALR.construir();
63     this.ventanaPadre.getGramatica().setColeccionLALR(
64         colLALR);
65     this.ventanaPadre.setColeccion(colLALR);
66 }
67 this.jTextArea1.setText(this.ventanaPadre.getGramatica
68     ().getColeccionLALR().getColeccion());
69 }
70 }
71 }
72 /**
73  * This method is called from within the constructor to initialize
74  * the form.

```

```

65      * WARNING: Do NOT modify this code. The content of this method is
        always
66      * regenerated by the Form Editor.
67      */
68      @SuppressWarnings("unchecked")
69      // <editor-fold defaultstate="collapsed" desc="Generated Code">//
        GEN-BEGIN: initComponents
70      private void initComponents() {
71
72          jLabel1 = new javax.swing.JLabel();
73          jScrollPane1 = new javax.swing.JScrollPane();
74          jTextArea1 = new javax.swing.JTextArea();
75          jButton1 = new javax.swing.JButton();
76          jButton2 = new javax.swing.JButton();
77          jButton3 = new javax.swing.JButton();
78          jButton4 = new javax.swing.JButton();
79          jButton5 = new javax.swing.JButton();
80
81          setBackground(new java.awt.Color(233, 244, 244));
82          setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
83
84          jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
85          jLabel1.setText("Coleccion Canonica Elementos LR(0)");
86
87          jTextArea1.setColumns(20);
88          jTextArea1.setRows(5);
89          jScrollPane1.setViewportView(jTextArea1);
90
91          jButton1.setIcon(new javax.swing.ImageIcon(getClass().
            getResource("/es/uco/simas/resources/ultimo.png"))); //
            NOI18N
92          jButton1.addActionListener(new java.awt.event.ActionListener()
            {
93              public void actionPerformed(java.awt.event.ActionEvent evt)
            {
94                  jButton1ActionPerformed(evt);
95              }
96          });
97
98          jButton2.setIcon(new javax.swing.ImageIcon(getClass().
            getResource("/es/uco/simas/resources/siguiente.png"))); //
            NOI18N
99          jButton2.addActionListener(new java.awt.event.ActionListener()
            {
100              public void actionPerformed(java.awt.event.ActionEvent evt)
            {
101                  jButton2ActionPerformed(evt);
102              }
103          });
104
105          jButton3.setIcon(new javax.swing.ImageIcon(getClass().
            getResource("/es/uco/simas/resources/anterior.png"))); //
            NOI18N

```

```

106     jButton3.addActionListener(new java.awt.event.ActionListener()
107     {
108         public void actionPerformed(java.awt.event.ActionEvent evt)
109         {
110             jButton3ActionPerformed(evt);
111         }
112     });
113
114     jButton4.setIcon(new javax.swing.ImageIcon(getClass().
115         getResource("/es/uco/simas/resources/primero.png"))); //
116         NOI18N
117     jButton4.addActionListener(new java.awt.event.ActionListener()
118     {
119         public void actionPerformed(java.awt.event.ActionEvent evt)
120         {
121             jButton4ActionPerformed(evt);
122         }
123     });
124
125     jButton5.setText(" Cancelar");
126     jButton5.addActionListener(new java.awt.event.ActionListener()
127     {
128         public void actionPerformed(java.awt.event.ActionEvent evt)
129         {
130             jButton5ActionPerformed(evt);
131         }
132     });
133
134     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
135         this);
136     this.setLayout(layout);
137     layout.setHorizontalGroup(
138         layout.createParallelGroup(javax.swing.GroupLayout.
139             Alignment.LEADING)
140         .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
141             layout.createSequentialGroup()
142                 .addGap(46, 46, 46)
143                 .addGroup(layout.createParallelGroup(javax.swing.
144                     GroupLayout.Alignment.TRAILING)
145                     .addComponent(jScrollPane1)
146                     .addGroup(layout.createSequentialGroup()
147                         .addComponent(jButton5)
148                         .addPreferredGap(javax.swing.LayoutStyle.
149                             ComponentPlacement.RELATED, javax.swing.
150                             GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
149                         .addComponent(jButton4)
150                         .addPreferredGap(javax.swing.LayoutStyle.
151                             ComponentPlacement.RELATED)
152                         .addComponent(jButton3)
153                         .addPreferredGap(javax.swing.LayoutStyle.
154                             ComponentPlacement.RELATED)
155                         .addComponent(jButton2))
156                     )
157             )
158     );

```

```

142         .addPreferredGap(javax.swing.LayoutStyle.
143             ComponentPlacement.RELATED)
144         .addComponent(jButton1))
145     .addGap(30, 30, 30))
146     .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
147         layout.createSequentialGroup()
148         .addContainerGap(201, Short.MAX_VALUE)
149         .addComponent(jLabel1)
150         .addGap(177, 177, 177))
151 );
152 layout.setVerticalGroup(
153     layout.createParallelGroup(javax.swing.GroupLayout.
154         Alignment.LEADING)
155     .addGroup(layout.createSequentialGroup()
156         .addGap(22, 22, 22)
157         .addComponent(jLabel1)
158         .addGap(35, 35, 35)
159         .addComponent(jScrollPane1, javax.swing.GroupLayout.
160             DEFAULT_SIZE, 502, Short.MAX_VALUE)
161         .addGap(18, 18, 18)
162         .addGroup(layout.createParallelGroup(javax.swing.
163             GroupLayout.Alignment.TRAILING)
164             .addGroup(layout.createParallelGroup(javax.swing.
165                 GroupLayout.Alignment.LEADING)
166                 .addComponent(jButton5)
167                 .addComponent(jButton4)
168                 .addComponent(jButton3)
169                 .addComponent(jButton2))
170             .addGroup(layout.createSequentialGroup()
171                 .addComponent(jButton1)
172                 .addGap(2, 2, 2)))
173         .addGap(19, 19, 19))
174 );
175 }// </editor-fold> //GEN-END: initComponents
176
177 private void jButton5ActionPerformed(java.awt.event.ActionEvent evt
178 ) { //GEN-FIRST: event_jButton5ActionPerformed
179     int conf = JOptionPane.showConfirmDialog(null, "¿Desea salir
180     del asistente de la simulacion de la gramática?", "Salir",
181     JOptionPane.YES_NO_OPTION);
182
183     if (conf == 0)
184         this.ventanaPadre.dispose();
185 } //GEN-LAST: event_jButton5ActionPerformed
186
187 private void jButton3ActionPerformed(java.awt.event.ActionEvent evt
188 ) { //GEN-FIRST: event_jButton3ActionPerformed
189     this.ventanaPadre.cambiarPaso(2);
190 } //GEN-LAST: event_jButton3ActionPerformed
191
192 private void jButton4ActionPerformed(java.awt.event.ActionEvent evt
193 ) { //GEN-FIRST: event_jButton4ActionPerformed
194     this.ventanaPadre.cambiarPaso(1);
195 } //GEN-LAST: event_jButton4ActionPerformed

```

```

184     } //GEN-LAST: event_jButton4ActionPerformed
185
186     private void jButton2ActionPerformed(java.awt.event.ActionEvent evt
187     ) { //GEN-FIRST: event_jButton2ActionPerformed
188         this.ventanaPadre.cambiarPaso(4);
189     } //GEN-LAST: event_jButton2ActionPerformed
190
191     private void jButton1ActionPerformed(java.awt.event.ActionEvent evt
192     ) { //GEN-FIRST: event_jButton1ActionPerformed
193
194         PanelNuevaSimAscPaso4 paso4 = new PanelNuevaSimAscPaso4(this.
195             ventanaPadre);
196
197         this.ventanaPadre.cambiarPaso(5);
198     } //GEN-LAST: event_jButton1ActionPerformed
199
200     // Variables declaration - do not modify //GEN-BEGIN: variables
201     private javax.swing.JButton jButton1;
202     private javax.swing.JButton jButton2;
203     private javax.swing.JButton jButton3;
204     private javax.swing.JButton jButton4;
205     private javax.swing.JButton jButton5;
206     private javax.swing.JLabel jLabel1;
207     private javax.swing.JScrollPane jScrollPane1;
208     private javax.swing.JTextArea jTextArea1;
209     // End of variables declaration //GEN-END: variables
210 }

```

### 2.5.9. PanelNuevaSimAscPaso4.java

```

1 //SimAS / Simulador
2 //Panel Nueva Simulacion Ascendente paso 4
3
4 package es.uco.simas.simulador;
5
6 import es.uco.simas.editor.TablaLR;
7 import es.uco.simas.util.gramatica.Gramatica;
8 import javax.swing.JOptionPane;
9
10 /**
11  * @author vanesa
12  */
13 public class PanelNuevaSimAscPaso4 extends javax.swing.JPanel {
14     private Gramatica gramatica;
15     private VentanaSimuladorAsc ventanaPadre;
16
17     public PanelNuevaSimAscPaso4(VentanaSimuladorAsc ventanaPadre) {
18         initComponents();
19         this.ventanaPadre = ventanaPadre;
20         this.gramatica = ventanaPadre.getGramatica();

```

```

21         this.jTable1.setDefaultRenderer (Object.class, new MiRender());
22         this.jTable2.setDefaultRenderer (Object.class, new MiRender());
23         if (this.ventanaPadre.getMetodo() == 0){
24             this.construirTablaSLR();
25         }
26         if (this.ventanaPadre.getMetodo() == 1){
27             this.construirTablaLR();
28         }
29         if (this.ventanaPadre.getMetodo() == 2){
30             this.construirTablaLALR();
31         }
32     }
33
34     public void construirTablaSLR(){
35         this.gramatica.generarTLR(1);
36         TablaLR tabla = this.gramatica.getTlr();
37
38         this.jTable1.setModel(tabla.getTAccion().getMatrizAccion());
39         this.jTable2.setModel(tabla.getTlrA().getMatrizIrA());
40
41         this.ventanaPadre.setGramatica(gramatica);
42     }
43
44     public void construirTablaLR(){
45         this.gramatica.generarTLR(2);
46         TablaLR tabla = this.gramatica.getTlr();
47
48         this.jTable1.setModel(tabla.getTAccion().getMatrizAccion());
49         this.jTable2.setModel(tabla.getTlrA().getMatrizIrA());
50
51         this.ventanaPadre.setGramatica(gramatica);
52     }
53
54     public void construirTablaLALR(){
55         this.gramatica.generarTLR(3);
56         TablaLR tabla = this.gramatica.getTlr();
57
58         this.jTable1.setModel(tabla.getTAccion().getMatrizAccion());
59         this.jTable2.setModel(tabla.getTlrA().getMatrizIrA());
60
61         this.ventanaPadre.setGramatica(gramatica);
62     }
63
64     /**
65      * This method is called from within the constructor to initialize
66      * the form.
67      * WARNING: Do NOT modify this code. The content of this method is
68      * always
69      * regenerated by the Form Editor.
70      */
71     @SuppressWarnings("unchecked")
72     // <editor-fold defaultstate="collapsed" desc="Generated Code">
73     GEN-BEGIN: initComponents

```

```

71     private void initComponents() {
72
73         jButton1 = new javax.swing.JButton();
74         jButton2 = new javax.swing.JButton();
75         jButton3 = new javax.swing.JButton();
76         jButton4 = new javax.swing.JButton();
77         jButton5 = new javax.swing.JButton();
78         jLabel1 = new javax.swing.JLabel();
79         jScrollPane1 = new javax.swing.JScrollPane();
80         jTable1 = new javax.swing.JTable();
81         jScrollPane2 = new javax.swing.JScrollPane();
82         jTable2 = new javax.swing.JTable();
83         jLabel2 = new javax.swing.JLabel();
84         jLabel3 = new javax.swing.JLabel();
85
86         setBackground(new java.awt.Color(233, 244, 244));
87
88         jButton1.setIcon(new javax.swing.ImageIcon(getClass().
            getResource("/es/uco/simas/resources/ultimo.png"))); //
            NOI18N
89         jButton1.addActionListener(new java.awt.event.ActionListener()
90         {
91             public void actionPerformed(java.awt.event.ActionEvent evt)
92             {
93                 jButton1ActionPerformed(evt);
94             }
95         });
96
97         jButton2.setIcon(new javax.swing.ImageIcon(getClass().
98             getResource("/es/uco/simas/resources/siguiente.png"))); //
99             NOI18N
100        jButton2.addActionListener(new java.awt.event.ActionListener()
101        {
102            public void actionPerformed(java.awt.event.ActionEvent evt)
103            {
104                jButton2ActionPerformed(evt);
105            }
106        });
107
108        jButton3.setIcon(new javax.swing.ImageIcon(getClass().
109            getResource("/es/uco/simas/resources/anterior.png"))); //
            NOI18N

```

```

110     jButton4.addActionListener(new java.awt.event.ActionListener()
111     {
112         public void actionPerformed(java.awt.event.ActionEvent evt)
113         {
114             jButton4ActionPerformed(evt);
115         }
116     });
117
118     jButton5.setText("Cancelar");
119     jButton5.addActionListener(new java.awt.event.ActionListener()
120     {
121         public void actionPerformed(java.awt.event.ActionEvent evt)
122         {
123             jButton5ActionPerformed(evt);
124         }
125     });
126
127     jLabel1.setFont(new java.awt.Font("Tahoma", 1, 36)); // NOI18N
128     jLabel1.setText("Tabla LR");
129
130     jTable1.setModel(new javax.swing.table.DefaultTableModel(
131         new Object [][] {
132             {null, null},
133             {null, null}
134         },
135         new String [] {
136             "", ""
137         }
138     ) {
139         boolean[] canEdit = new boolean [] {
140             false, false
141         };
142
143         public boolean isCellEditable(int rowIndex, int columnIndex
144         ) {
145             return canEdit [columnIndex];
146         }
147     });
148     jScrollPane1.setViewportViewView(jTable1);
149
150     jTable2.setModel(new javax.swing.table.DefaultTableModel(
151         new Object [][] {
152             {null, null, null, null},
153             {null, null, null, null},
154             {null, null, null, null},
155             {null, null, null, null}
156         },
157         new String [] {
158             "", "", "", ""
159         }
160     ) {
161         boolean[] canEdit = new boolean [] {
162             false, false, false, false

```



```

158         };
159
160         public boolean isCellEditable(int rowIndex, int columnIndex
161         ) {
162             return canEdit [columnIndex];
163         }
164     });
165     jScrollPane2.setViewportViewView(jTable2);
166
167     jLabel2.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
168     jLabel2.setText("Parte Accion");
169
170     jLabel3.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
171     jLabel3.setText("Parte Ir_a");
172
173     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
174     this);
175     this.setLayout(layout);
176     layout.setHorizontalGroup(
177     layout.createParallelGroup(javax.swing.GroupLayout.
178     Alignment.LEADING)
179     .addGroup(layout.createSequentialGroup()
180     .addComponent(jButton5, javax.swing.GroupLayout.
181     PREFERRED_SIZE, 92, javax.swing.
182     GroupLayout.PREFERRED_SIZE)
183     .addPreferredGap(javax.swing.LayoutStyle.
184     ComponentPlacement.RELATED, javax.swing.
185     GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
186     .addComponent(jButton4)
187     .addPreferredGap(javax.swing.LayoutStyle.
188     ComponentPlacement.UNRELATED)
189     .addComponent(jButton3)
190     .addPreferredGap(javax.swing.LayoutStyle.
191     ComponentPlacement.UNRELATED)
192     .addComponent(jButton2)
193     .addPreferredGap(javax.swing.LayoutStyle.
194     ComponentPlacement.RELATED)
195     .addComponent(jButton1)
196     .addGap(20, 20, 20))
197     .addGroup(layout.createSequentialGroup()
198     .addComponent(jScrollPane1)
199     .addPreferredGap(javax.swing.LayoutStyle.
200     ComponentPlacement.RELATED)
201     .addComponent(jScrollPane2, javax.swing.
202     GroupLayout.PREFERRED_SIZE, 323, javax.
203     swing.GroupLayout.PREFERRED_SIZE)
204     .addGap(32, 32, 32)))
205     .addGroup(layout.createSequentialGroup()
206     .addGap(307, 307, 307)

```

```

197         .addComponent(jLabel1)
198         .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
199             Short.MAX_VALUE))
200     .addGroup(layout.createSequentialGroup())
201     .addGap(94, 94, 94)
202     .addComponent(jLabel2)
203     .addPreferredGap(javax.swing.LayoutStyle.
204         ComponentPlacement.RELATED, 415, Short.MAX_VALUE)
205     .addComponent(jLabel3)
206     .addGap(176, 176, 176))
207 );
208 layout.setVerticalGroup(
209     layout.createParallelGroup(javax.swing.GroupLayout.
210         Alignment.LEADING)
211     .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
212         layout.createSequentialGroup()
213         .addGroup(layout.createParallelGroup(javax.swing.
214             GroupLayout.Alignment.LEADING)
215             .addGroup(layout.createSequentialGroup()
216                 .addGap(73, 73, 73)
217                 .addGroup(layout.createParallelGroup(javax.
218                     swing.GroupLayout.Alignment.BASELINE)
219                     .addComponent(jLabel2)
220                     .addComponent(jLabel3)))
221             .addGroup(layout.createSequentialGroup()
222                 .addContainerGap()
223                 .addComponent(jLabel1)))
224         .addGap(18, 18, 18)
225         .addGroup(layout.createParallelGroup(javax.swing.
226             GroupLayout.Alignment.LEADING)
227             .addComponent(jScrollPane2, javax.swing.GroupLayout.
228                 DEFAULT_SIZE, 408, Short.MAX_VALUE)
229             .addComponent(jScrollPane1, javax.swing.GroupLayout.
230                 PREFERRED_SIZE, 0, Short.MAX_VALUE))
231         .addGroup(layout.createParallelGroup(javax.swing.
232             GroupLayout.Alignment.LEADING)
233             .addGroup(layout.createSequentialGroup()
234                 .addGap(18, 18, 18)
235                 .addGroup(layout.createParallelGroup(javax.
236                     swing.GroupLayout.Alignment.BASELINE)
237                     .addComponent(jButton1)
238                     .addComponent(jButton2)
239                     .addComponent(jButton3)
240                     .addComponent(jButton4))
241                 .addGap(31, 31, 31))
242             .addGroup(javax.swing.GroupLayout.Alignment.
243                 TRAILING, layout.createSequentialGroup()
244                 .addPreferredGap(javax.swing.LayoutStyle.
245                     ComponentPlacement.RELATED)
246                 .addComponent(jButton5)
247                 .addGap(28, 28, 28)))
248     );
249 } // </editor-fold> //GEN-END: initComponents

```

```

237
238     private void jButton3ActionPerformed(java.awt.event.ActionEvent evt
239         ) {//GEN-FIRST:event_jButton3ActionPerformed
240         this.ventanaPadre.cambiarPaso(3);
241     }//GEN-LAST:event_jButton3ActionPerformed
242
243     private void jButton4ActionPerformed(java.awt.event.ActionEvent evt
244         ) {//GEN-FIRST:event_jButton4ActionPerformed
245         this.ventanaPadre.cambiarPaso(1);
246     }//GEN-LAST:event_jButton4ActionPerformed
247
248     private void jButton5ActionPerformed(java.awt.event.ActionEvent evt
249         ) {//GEN-FIRST:event_jButton5ActionPerformed
250         int conf = JOptionPane.showConfirmDialog(null, "¿Desea salir
251         del asistente de la simulacion de la gramática?", "Salir",
252         JOptionPane.YES_NO_OPTION);
253
254         if (conf==0)
255             this.ventanaPadre.dispose();
256     }//GEN-LAST:event_jButton5ActionPerformed
257
258     private void jButton2ActionPerformed(java.awt.event.ActionEvent evt
259         ) {//GEN-FIRST:event_jButton2ActionPerformed
260         this.ventanaPadre.cambiarPaso(5);
261     }//GEN-LAST:event_jButton2ActionPerformed
262
263     private void jButton1ActionPerformed(java.awt.event.ActionEvent evt
264         ) {//GEN-FIRST:event_jButton1ActionPerformed
265         this.ventanaPadre.cambiarPaso(5);
266     }//GEN-LAST:event_jButton1ActionPerformed
267
268     // Variables declaration - do not modify//GEN-BEGIN:variables
269     private javax.swing.JButton jButton1;
270     private javax.swing.JButton jButton2;
271     private javax.swing.JButton jButton3;
272     private javax.swing.JButton jButton4;
273     private javax.swing.JButton jButton5;
274     private javax.swing.JLabel jLabel1;
275     private javax.swing.JLabel jLabel2;
276     private javax.swing.JLabel jLabel3;
277     private javax.swing.JScrollPane jScrollPane1;
278     private javax.swing.JScrollPane jScrollPane2;
279     private javax.swing.JTable jTable1;
280     private javax.swing.JTable jTable2;
281     // End of variables declaration//GEN-END:variables

```

### 2.5.10. PanelNuevaSimAscPaso5.java

```

1 //SimAS / Simulador
2 //Panel Nueva Simulacion Ascendente paso 5
3
4 package es.uco.simas.simulador;
5
6 import es.uco.simas.editor.FuncionError;
7 import es.uco.simas.util.gramatica.Gramatica;
8 import java.util.ArrayList;
9 import javax.swing.DefaultListModel;
10 import javax.swing.JOptionPane;
11
12 /**
13  * @author Vanesa
14  */
15 public class PanelNuevaSimAscPaso5 extends javax.swing.JPanel {
16
17     private VentanaSimuladorAsc ventanaPadre;
18     private Gramatica gramatica;
19     private int funError=0;
20
21     public PanelNuevaSimAscPaso5(VentanaSimuladorAsc ventanaPadre) {
22         initComponents();
23         this.ventanaPadre = ventanaPadre;
24         this.gramatica = ventanaPadre.getGramatica();
25
26         if(this.gramatica.getTlr().getTAccion().getFunError().size() !=
27             0){
28             this.funcionError();
29             this.funError = 1;
30         }
31         if(funError ==0){
32             this.jButtonFinalizar.setVisible(true);
33             this.jButtonUltimo.setVisible(false);
34             this.jButtonSiguiente.setEnabled(false);
35             this.jButtonEliminar.setEnabled(false);
36         }
37     }
38
39     /**
40      * This method is called from within the constructor to initialize
41      * the form.
42      * WARNING: Do NOT modify this code. The content of this method is
43      * always
44      * regenerated by the Form Editor.
45      */
46     @SuppressWarnings("unchecked")
47     // <editor-fold defaultstate="collapsed" desc="Generated Code">
48     GEN-BEGIN: initComponents {
49         jLabel1 = new javax.swing.JLabel();
50         jButtonCancelar = new javax.swing.JButton();

```

```

49     jButtonUltimo = new javax.swing.JButton();
50     jButtonSiguiente = new javax.swing.JButton();
51     jButton3 = new javax.swing.JButton();
52     jButton4 = new javax.swing.JButton();
53     jScrollPane1 = new javax.swing.JScrollPane();
54     jList1 = new javax.swing.JList();
55     jCheckBox1 = new javax.swing.JCheckBox();
56     jButtonNueva = new javax.swing.JButton();
57     jButtonEliminar = new javax.swing.JButton();
58     jButtonFinalizar = new javax.swing.JButton();
59
60     setBackground(new java.awt.Color(233, 244, 244));
61
62     jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
63     jLabel1.setText("Funciones de Error");
64
65     jButtonCancelar.setText("Cancelar");
66     jButtonCancelar.addActionListener(new java.awt.event.
        ActionListener() {
67         public void actionPerformed(java.awt.event.ActionEvent evt)
68         {
69             jButtonCancelarActionPerformed(evt);
70         }
71     });
72     jButtonUltimo.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/ultimo.png"))); //
        NOI18N
73     jButtonUltimo.addActionListener(new java.awt.event.
        ActionListener() {
74         public void actionPerformed(java.awt.event.ActionEvent evt)
75         {
76             jButtonUltimoActionPerformed(evt);
77         }
78     });
79     jButtonSiguiente.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/siguiente.png"))); //
        NOI18N
80     jButtonSiguiente.addActionListener(new java.awt.event.
        ActionListener() {
81         public void actionPerformed(java.awt.event.ActionEvent evt)
82         {
83             jButtonSiguienteActionPerformed(evt);
84         }
85     });
86     jButton3.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/anterior.png"))); //
        NOI18N
87     jButton3.addActionListener(new java.awt.event.ActionListener()
        {

```

```

88         public void actionPerformed(java.awt.event.ActionEvent evt)
89         {
90             jButton3ActionPerformed(evt);
91         }
92     });
93     jButton4.setIcon(new javax.swing.ImageIcon(getClass().
94         getResource("/es/uco/simas/resources/primer0.png"))); //
95         NOI18N
96     jButton4.addActionListener(new java.awt.event.ActionListener()
97     {
98         public void actionPerformed(java.awt.event.ActionEvent evt)
99         {
100             jButton4ActionPerformed(evt);
101         }
102     });
103     jScrollPane1.setViewportView(jList1);
104
105     jCheckBox1.setText("No utilizar Funciones de error");
106     jCheckBox1.addActionListener(new java.awt.event.ActionListener
107     () {
108         public void actionPerformed(java.awt.event.ActionEvent evt)
109         {
110             jCheckBox1ActionPerformed1(evt);
111         }
112     });
113
114     jButtonNueva.setText("Nueva Funcion Error ");
115     jButtonNueva.addActionListener(new java.awt.event.
116         ActionListener() {
117         public void actionPerformed(java.awt.event.ActionEvent evt)
118         {
119             jButtonNuevaActionPerformed(evt);
120         }
121     });
122
123     jButtonEliminar.setText("Eliminar Funcion Error");
124     jButtonEliminar.addActionListener(new java.awt.event.
125         ActionListener() {
126         public void actionPerformed(java.awt.event.ActionEvent evt)
127         {

```

```

127         jButtonFinalizarActionPerformed(evt);
128     }
129 });
130
131 javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
132     this);
133 this.setLayout(layout);
134 layout.setHorizontalGroup(
135     layout.createParallelGroup(javax.swing.GroupLayout.
136         Alignment.LEADING)
137     .addGroup(layout.createSequentialGroup()
138         .addGap(21, 21, 21)
139         .addGroup(layout.createParallelGroup(javax.swing.
140             GroupLayout.Alignment.LEADING)
141             .addGroup(layout.createSequentialGroup()
142                 .addComponent(jButtonCancelar)
143                 .addPreferredGap(javax.swing.LayoutStyle.
144                     ComponentPlacement.RELATED, javax.swing.
145                         GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
146                 .addComponent(jButton4)
147                 .addPreferredGap(javax.swing.LayoutStyle.
148                     ComponentPlacement.UNRELATED)
149                 .addComponent(jButton3)
150                 .addPreferredGap(javax.swing.LayoutStyle.
151                     ComponentPlacement.RELATED)
152                 .addComponent(jButtonSiguiente)
153                 .addPreferredGap(javax.swing.LayoutStyle.
154                     ComponentPlacement.RELATED)
155                 .addComponent(jButtonUltimo)
156                 .addPreferredGap(javax.swing.LayoutStyle.
157                     ComponentPlacement.RELATED)
158                 .addComponent(jButtonFinalizar))
159             .addGroup(layout.createSequentialGroup()
160                 .addComponent(jScrollPane1, javax.swing.
161                     GroupLayout.DEFAULT_SIZE, 472, Short.
162                         MAX_VALUE)
163                 .addGap(18, 18, 18)
164                 .addGroup(layout.createParallelGroup(javax.
165                     swing.GroupLayout.Alignment.LEADING)
166                     .addComponent(jButtonNueva, javax.swing.
167                         GroupLayout.Alignment.TRAILING)
168                     .addComponent(jButtonEliminar, javax.swing.
169                         GroupLayout.Alignment.TRAILING))))
170         .addGap(24, 24, 24))
171     .addGroup(layout.createSequentialGroup()
172         .addGroup(layout.createParallelGroup(javax.swing.
173             GroupLayout.Alignment.LEADING)
174             .addGroup(layout.createSequentialGroup()
175                 .addGap(48, 48, 48)
176                 .addComponent(jCheckBox1))
177             .addGroup(layout.createSequentialGroup()
178                 .addGap(205, 205, 205)
179                 .addComponent(jLabel1)))
180         .addContainerGap(10, Short.MAX_VALUE))
181 );

```

```

165         .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
166         );
167     layout.setVerticalGroup(
168         layout.createParallelGroup(javax.swing.GroupLayout.
169             Alignment.LEADING)
170         .addGroup(layout.createSequentialGroup()
171             .addGroup(layout.createParallelGroup(javax.swing.
172                 GroupLayout.Alignment.LEADING)
173                 .addGroup(layout.createSequentialGroup()
174                     .addGap(102, 102, 102)
175                     .addComponent(jButtonNueva)
176                     .addGap(41, 41, 41)
177                     .addComponent(jButtonEliminar)
178                     .addPreferredGap(javax.swing.LayoutStyle.
179                         ComponentPlacement.RELATED, 183, Short.
180                             MAX_VALUE))
181                     .addGroup(layout.createSequentialGroup()
182                         .addGap(22, 22, 22)
183                         .addComponent(jLabel1)
184                         .addPreferredGap(javax.swing.LayoutStyle.
185                             ComponentPlacement.RELATED)
186                         .addComponent(jScrollPane1)
187                         .addPreferredGap(javax.swing.LayoutStyle.
188                             ComponentPlacement.UNRELATED)))
189                     .addComponent(jCheckBox1)
190                     .addGap(35, 35, 35)
191                     .addGroup(layout.createParallelGroup(javax.swing.
192                         GroupLayout.Alignment.LEADING)
193                         .addComponent(jButtonFinalizar)
194                         .addComponent(jButtonUltimo)
195                         .addComponent(jButton3)
196                         .addComponent(jButtonCancelar)
197                         .addComponent(jButton4)
198                         .addComponent(jButtonSiguiente))
199                     .addGap(19, 19, 19))
200         );
201 }// </editor-fold>//GEN-END: initComponents
202
203 public void funcionError() {
204     DefaultListModel lista = new DefaultListModel();
205     ArrayList<FuncionError> funError = this.gramatica.getTlr().
206         getTAccion().getFunError();
207     StringBuilder string = new StringBuilder();
208     int i=0;
209
210     while(i < funError.size()){
211         string = new StringBuilder();
212         int accion;
213         string = string.append(funError.get(i).getIdentificador());
214         string = string.append(" - ");
215         accion = funError.get(i).getAccion();
216         if(accion == 1)

```



```

209         string.append("Insertar un S mbolo en la Entrada: ");
210     if(accion == 2)
211         string.append("Borrar un S mbolo de la Entrada");
212     if(accion == 3)
213         string.append("Modificar un S mbolo de la Entrada: ");
214     if(accion == 4)
215         string.append("Insertar un S mbolo de la Pila: ");
216     if(accion == 5)
217         string.append("Borrar un S mbolo de la Pila");
218     if(accion == 6)
219         string.append("Modificar un S mbolo de la Pila: ");
220     if(accion == 7)
221         string.append("Terminar el an lisis");
222     if(accion == 1 || accion == 3 || accion == 4 || accion == 6)
223         string.append(funError.get(i).getSimbolo().getNombre());
224         ;
225     lista.add(i, string);
226     i++;
227
228 }
229 this.jList1.setModel(lista);
230
231 if(this.jList1.getModel().getSize() != 0){
232     this.jButtonFinalizar.setVisible(false);
233     this.jButtonUltimo.setVisible(true);
234     this.jButtonSiguiente.setEnabled(true);
235     this.jButtonNueva.setEnabled(true);
236     this.jButtonEliminar.setEnabled(true);
237 }else{
238     this.jButtonNueva.setEnabled(true);
239     this.jButtonEliminar.setEnabled(false);
240     this.jButtonFinalizar.setVisible(true);
241     this.jButtonUltimo.setVisible(false);
242     this.jButtonSiguiente.setEnabled(false);
243 }
244
245 }
246 void setGramatica(Gramatica gr){
247     this.ventanaPadre.setGramatica(gr);
248     this.gramatica = gr;
249 }
250 private void jButtonCancelarActionPerformed(java.awt.event.
ActionEvent evt) { //GEN-FIRST:
event_jButtonCancelarActionPerformed
251     int conf = JOptionPane.showConfirmDialog(null, "  Desea salir
del asistente de la simulacion de la gram tica?", "Salir",
JOptionPane.YES_NO_OPTION);
252
253     if(conf == 0)
254         this.ventanaPadre.dispose();
255 } //GEN-LAST: event_jButtonCancelarActionPerformed

```

```

256
257     private void jButton4ActionPerformed(java.awt.event.ActionEvent evt
258         ) {//GEN-FIRST: event_jButton4ActionPerformed
259         this.ventanaPadre.cambiarPaso(1);
260     }//GEN-LAST: event_jButton4ActionPerformed
261
262     private void jButton3ActionPerformed(java.awt.event.ActionEvent evt
263         ) {//GEN-FIRST: event_jButton3ActionPerformed
264         this.ventanaPadre.cambiarPaso(3);
265     }//GEN-LAST: event_jButton3ActionPerformed
266
267     private void jButtonSiguienteActionPerformed(java.awt.event.
268         ActionEvent evt) {//GEN-FIRST:
269         event_jButtonSiguienteActionPerformed
270         this.ventanaPadre.cambiarPaso(6);
271     }//GEN-LAST: event_jButtonSiguienteActionPerformed
272
273     private void jButtonNuevaActionPerformed(java.awt.event.ActionEvent
274         evt) {//GEN-FIRST: event_jButtonNuevaActionPerformed
275         NuevaFuncionError error = new NuevaFuncionError(this.gramatica ,
276             this, 1);
277         error.setLocationRelativeTo(null);
278         error.setVisible(true);
279     }//GEN-LAST: event_jButtonNuevaActionPerformed
280
281     private void jButton1ActionPerformed1(java.awt.event.ActionEvent
282         evt) {//GEN-FIRST: event_jButton1ActionPerformed1
283         if(this.jCheckBox1.isSelected()){
284             this.jButtonNueva.setEnabled(false);
285             this.jButtonEliminar.setEnabled(false);
286             this.jButtonFinalizar.setVisible(true);
287             this.jButtonUltimo.setVisible(false);
288             this.jButtonSiguiente.setEnabled(false);
289         } else {
290             if(this.jList1.getModel().getSize()!=0){
291                 this.jButtonFinalizar.setVisible(false);
292                 this.jButtonUltimo.setVisible(true);
293                 this.jButtonSiguiente.setEnabled(true);
294                 this.jButtonNueva.setEnabled(true);
295                 this.jButtonEliminar.setEnabled(true);
296             } else {
297                 this.jButtonNueva.setEnabled(true);
298                 this.jButtonEliminar.setEnabled(false);
299                 this.jButtonFinalizar.setVisible(true);
300                 this.jButtonUltimo.setVisible(false);
301                 this.jButtonSiguiente.setEnabled(false);
302             }
303         }
304     }//GEN-LAST: event_jButton1ActionPerformed1
305
306     private void jButtonFinalizarActionPerformed(java.awt.event.
307         ActionEvent evt) {//GEN-FIRST:
308         event_jButtonFinalizarActionPerformed

```

```

300      /*Simulador sim = new Simulador(2, this.gramatica);
301      sim.setVisible(true);
302      sim.setLocationRelativeTo(null);*/
303      this.ventanaPadre.finalizarAsistente();
304      this.ventanaPadre.dispose();
305  } //GEN-LAST: event_jButtonFinalizarActionPerformed
306
307  private void jButtonEliminarActionPerformed(java.awt.event.
    ActionEvent evt) {//GEN-FIRST:
    event_jButtonEliminarActionPerformed
308      DefaultListModel modelo= null;
309      int seleccion= this.jList1.getSelectedIndex();
310      String funcion = this.jList1.getModel().getElementAt(seleccion)
        .toString();
311      ArrayList<FuncionError> funError = this.gramatica.getTlr().
        getTAccion().getFunError();
312      String id = funcion.substring(0, 1);
313      int num = Integer.parseInt(id);
314      int i=0;
315
316      if(seleccion != -1) {
317          modelo=(DefaultListModel) this.jList1.getModel();
318          modelo.remove(seleccion);
319
320          while(i < funError.size()){
321              if(funError.get(i).getIdentificador() == num){
322                  funError.remove(i);
323              } else
324                  i++;
325          }
326          this.gramatica.getTlr().getTAccion().setFunError(funError);
327      }
328
329      if(this.jList1.getModel().getSize()!=0){
330          this.jButtonFinalizar.setVisible(false);
331          this.jButtonUltimo.setVisible(true);
332          this.jButtonSiguiente.setEnabled(true);
333          this.jButtonNueva.setEnabled(true);
334          this.jButtonEliminar.setEnabled(true);
335      } else{
336          this.jButtonNueva.setEnabled(true);
337          this.jButtonEliminar.setEnabled(false);
338          this.jButtonFinalizar.setVisible(true);
339          this.jButtonUltimo.setVisible(false);
340          this.jButtonSiguiente.setEnabled(false);
341      }
342  } //GEN-LAST: event_jButtonEliminarActionPerformed
343
344  private void jButtonUltimoActionPerformed(java.awt.event.
    ActionEvent evt) {//GEN-FIRST:
    event_jButtonUltimoActionPerformed
345      this.ventanaPadre.cambiarPaso(6);
346  } //GEN-LAST: event_jButtonUltimoActionPerformed

```

```

347
348 // Variables declaration – do not modify//GEN-BEGIN:variables
349 private javax.swing.JButton jButton3;
350 private javax.swing.JButton jButton4;
351 private javax.swing.JButton jButtonCancelar;
352 private javax.swing.JButton jButtonEliminar;
353 private javax.swing.JButton jButtonFinalizar;
354 private javax.swing.JButton jButtonNueva;
355 private javax.swing.JButton jButtonSiguiente;
356 private javax.swing.JButton jButtonUltimo;
357 private javax.swing.JCheckBox jCheckBox1;
358 private javax.swing.JLabel jLabel1;
359 private javax.swing.JList jList1;
360 private javax.swing.JScrollPane jScrollPane1;
361 // End of variables declaration//GEN-END:variables
362 }

```

### 2.5.11. PanelNuevaSimAscPaso6.java

```

1 //SimAS / Simulador
2 //Panel Nueva Simulacion Ascendente paso 6
3
4 package es.uco.simas.simulador;
5
6 import es.uco.simas.editor.FuncionError;
7 import es.uco.simas.editor.ParteAccion;
8 import es.uco.simas.editor.TablaLR;
9 import es.uco.simas.util.gramatica.Gramatica;
10 import java.util.ArrayList;
11 import javax.swing.DefaultComboBoxModel;
12 import javax.swing.JOptionPane;
13 import javax.swing.table.DefaultTableModel;
14
15 /**
16  * @author Vanesa
17  */
18 public class PanelNuevaSimAscPaso6 extends javax.swing.JPanel {
19
20     private VentanaSimuladorAsc ventanaPadre;
21     private Gramatica gramatica;
22     private DefaultTableModel modeloConjuntos ;
23     private TablaLR tlr;
24     int red = 0;
25
26     public PanelNuevaSimAscPaso6(VentanaSimuladorAsc ventanaPadre) {
27         DefaultTableModel tabla = new DefaultTableModel();
28         this.modeloConjuntos = tabla;
29         initComponents();
30         this.ventanaPadre = ventanaPadre;
31         this.gramatica = ventanaPadre.getGramatica();

```

```

32     this.jButton6.setEnabled( false );
33     this.tlr = this.gramatica.getTlr();
34     this.jTable1.setDefaultRenderer ( Object.class , new MiRender());
35     this.jTable1.setModel( this.tlr.getTAccion().getMatrizAccion());
36     this.inicializarCombos( this.tlr );
37     this.jButton7.setText(" Completar con Reducciones");
38 }
39
40 /**
41  * This method is called from within the constructor to initialize
42  * the form.
43  * WARNING: Do NOT modify this code. The content of this method is
44  * always
45  * regenerated by the Form Editor.
46  */
47 // @SuppressWarnings("unchecked")
48 // <editor-fold defaultstate="collapsed" desc="Generated Code"> //
49 GEN-BEGIN: initComponents
50 private void initComponents() {
51
52     jLabel1 = new javax.swing.JLabel();
53     jButton1 = new javax.swing.JButton();
54     jButton2 = new javax.swing.JButton();
55     jButton3 = new javax.swing.JButton();
56     jButton4 = new javax.swing.JButton();
57     jScrollPane1 = new javax.swing.JScrollPane();
58     jTable1 = new javax.swing.JTable();
59     jButton5 = new javax.swing.JButton();
60     jComboBox1 = new javax.swing.JComboBox();
61     jLabel2 = new javax.swing.JLabel();
62     jButton6 = new javax.swing.JButton();
63     jButton7 = new javax.swing.JButton();
64
65     setBackground(new java.awt.Color(233, 244, 244));
66
67     jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
68     jLabel1.setText("Incluir Funciones de Error");
69
70     jButton1.setFont(new java.awt.Font("Tahoma", 1, 15)); // NOI18N
71     jButton1.setText(" Finalizar");
72     jButton1.addActionListener(new java.awt.event.ActionListener()
73     {
74         public void actionPerformed(java.awt.event.ActionEvent evt)
75         {
76             jButton1ActionPerformed(evt);
77         }
78     });
79
80     jButton2.setIcon(new javax.swing.ImageIcon(getClass().
81         getResource("/es/uco/simas/resources/siguiente.png"))); //
82     NOI18N
83     jButton2.setEnabled( false );

```

```

78     JButton3.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/anterior.png"))); //
        NOI18N
79     JButton3.addActionListener(new java.awt.event.ActionListener()
        {
80         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
81             JButton3ActionPerformed(evt);
82         }
83     });
84
85     JButton4.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/primer.png"))); //
        NOI18N
86     JButton4.addActionListener(new java.awt.event.ActionListener()
        {
87         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
88             JButton4ActionPerformed(evt);
89         }
90     });
91
92     jTable1.setModel(new javax.swing.table.DefaultTableModel(
93         new Object [][] {
94             {null, null, null, null},
95             {null, null, null, null},
96             {null, null, null, null},
97             {null, null, null, null}
98         },
99         new String [] {
100             "", "", "", ""
101         }
102     ) {
103         boolean[] canEdit = new boolean [] {
104             false, false, false, false
105         };
106
107         public boolean isCellEditable(int rowIndex, int columnIndex
            ) {
108             return canEdit [columnIndex];
109         }
110     });
111     jTable1.addMouseListener(new java.awt.event.MouseAdapter() {
112         public void mouseClicked(java.awt.event.MouseEvent evt) {
113             jTable1MouseClicked(evt);
114         }
115     });
116     jScrollPane1.setViewportViewView(jTable1);
117
118     JButton5.setFont(new java.awt.Font("Tahoma", 1, 15)); // NOI18N
119     JButton5.setText("Cancelar");
120     JButton5.addActionListener(new java.awt.event.ActionListener()
        {

```

```

121         public void actionPerformed(java.awt.event.ActionEvent evt)
122         {
123             jButton5ActionPerformed(evt);
124         }
125     });
126     jLabel2.setText("Pulse en la tabla para insertar la funcion de
127         error:");
128     jButton6.setText("Eliminar Funcion de Error");
129     jButton6.addActionListener(new java.awt.event.ActionListener()
130     {
131         public void actionPerformed(java.awt.event.ActionEvent evt)
132         {
133             jButton6ActionPerformed(evt);
134         }
135     });
136     jButton7.addActionListener(new java.awt.event.ActionListener()
137     {
138         public void actionPerformed(java.awt.event.ActionEvent evt)
139         {
140             jButton7ActionPerformed(evt);
141         }
142     });
143     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
144         this);
145     this.setLayout(layout);
146     layout.setHorizontalGroup(
147         layout.createParallelGroup(javax.swing.GroupLayout.
148             Alignment.LEADING)
149         .addGroup(layout.createSequentialGroup()
150             .addGroup(layout.createParallelGroup(javax.swing.
151                 GroupLayout.Alignment.LEADING)
152             .addGroup(layout.createSequentialGroup()
153                 .addGap(26, 26, 26)
154                 .addComponent(jButton5)
155                 .addGap(255, 255, 255)
156                 .addComponent(jButton4)
157                 .addPreferredGap(javax.swing.LayoutStyle.
158                     ComponentPlacement.RELATED)
159                 .addComponent(jButton3)
160                 .addPreferredGap(javax.swing.LayoutStyle.
161                     ComponentPlacement.RELATED)
162                 .addComponent(jButton2)
163                 .addPreferredGap(javax.swing.LayoutStyle.
164                     ComponentPlacement.RELATED)
165                 .addComponent(jButton1))
166             .addGroup(layout.createSequentialGroup()
167                 .addGap(70, 70, 70)
168                 .addComponent(jButton6)
169                 .addGap(61, 61, 61))
170         )
171     );

```

```

162         .addComponent(jButton7))
163     .addGroup(layout.createSequentialGroup())
164     .addGap(164, 164, 164)
165     .addComponent(jLabel1))
166     .addGroup(layout.createSequentialGroup())
167     .addGap(121, 121, 121)
168     .addComponent(jLabel2))
169     .addGroup(layout.createSequentialGroup())
170     .addGap(191, 191, 191)
171     .addComponent(jComboBox1, javax.swing.
        GroupLayout.PREFERRED_SIZE, javax.swing.
        GroupLayout.DEFAULT_SIZE, javax.swing.
        GroupLayout.PREFERRED_SIZE))
172     .addGroup(layout.createSequentialGroup())
173     .addContainerGap()
174     .addComponent(jScrollPane1, javax.swing.
        GroupLayout.PREFERRED_SIZE, 570, javax.
        swing.GroupLayout.PREFERRED_SIZE)))
175     .addContainerGap(28, Short.MAX_VALUE))
176 );
177 layout.setVerticalGroup(
178     layout.createParallelGroup(javax.swing.GroupLayout.
        Alignment.LEADING)
179     .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
        layout.createSequentialGroup()
180         .addGap(15, 15, 15)
181         .addComponent(jLabel1)
182         .addPreferredGap(javax.swing.LayoutStyle.
        ComponentPlacement.RELATED)
183         .addComponent(jLabel2)
184         .addGap(18, 18, 18)
185         .addComponent(jComboBox1, javax.swing.GroupLayout.
        PREFERRED_SIZE, javax.swing.GroupLayout.
        DEFAULT_SIZE, javax.swing.GroupLayout.
        PREFERRED_SIZE)
186         .addPreferredGap(javax.swing.LayoutStyle.
        ComponentPlacement.UNRELATED)
187         .addComponent(jScrollPane1, javax.swing.GroupLayout.
        DEFAULT_SIZE, 398, Short.MAX_VALUE)
188         .addPreferredGap(javax.swing.LayoutStyle.
        ComponentPlacement.UNRELATED)
189         .addGroup(layout.createParallelGroup(javax.swing.
        GroupLayout.Alignment.LEADING)
190             .addComponent(jButton6)
191             .addComponent(jButton7))
192         .addGap(26, 26, 26)
193         .addGroup(layout.createParallelGroup(javax.swing.
        GroupLayout.Alignment.LEADING)
194             .addComponent(jButton5)
195             .addGroup(layout.createSequentialGroup()
196                 .addGap(2, 2, 2)
197                 .addGroup(layout.createParallelGroup(javax.
                    swing.GroupLayout.Alignment.TRAILING)

```



```

198         .addComponent(jButton3)
199         .addComponent(jButton4)
200         .addComponent(jButton2)
201         .addComponent(jButton1)))
202     .addGap(23, 23, 23))
203 );
204 }// </editor-fold>//GEN-END: initComponents
205
206 private void inicializarCombos(TablaLR tlr) {
207     ArrayList<String> listaCombo = new ArrayList();
208     ArrayList<FuncionError> funError = tlr.getTAccion().getFunError
209     ();
210     StringBuilder string = new StringBuilder();
211     int i= 0;
212
213     while(i < funError.size()) {
214         string = new StringBuilder();
215         int accion;
216         string = string.append(funError.get(i).getIdentificador());
217         string = string.append(" - ");
218         accion = funError.get(i).getAccion();
219         if(accion == 1)
220             string.append("Insertar un S mbolo en la Entrada: ");
221         if(accion == 2)
222             string.append("Borrar un S mbolo de la Entrada");
223         if(accion == 3)
224             string.append("Modificar un S mbolo de la Entrada: ");
225         if(accion == 4)
226             string.append("Insertar un S mbolo de la Pila: ");
227         if(accion == 5)
228             string.append("Borrar un S mbolo de la Pila");
229         if(accion == 6)
230             string.append("Modificar un S mbolo de la Pila: ");
231         if(accion == 7)
232             string.append("Terminar el an lisis");
233         if(accion == 1 || accion == 3 || accion == 4 || accion == 6)
234             string.append(funError.get(i).getSimbolo().getNombre())
235             ;
236
237         listaCombo.add(string.toString());
238         i++;
239     }
240     DefaultComboBoxModel combo = new DefaultComboBoxModel(
241         listaCombo.toArray());
242     this.jComboBox1.setModel(combo);
243 }
244
245 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt
246 ) { //GEN-FIRST:event_jButton1ActionPerformed
247     this.gramatica.setTlr(this.tlr);
248     this.ventanaPadre.finalizarAsistente();
249     this.ventanaPadre.dispose();

```

```

246 } //GEN-LAST: event_jButton1ActionPerformed
247
248 private void jButton5ActionPerformed(java.awt.event.ActionEvent evt
249 ) { //GEN-FIRST: event_jButton5ActionPerformed
250     int conf = JOptionPane.showConfirmDialog(null, "¿Desea salir
251     del asistente de la simulacion de la gramática?", "Salir",
252     JOptionPane.YES_NO_OPTION);
253
254     if (conf==0)
255         this.ventanaPadre.dispose();
256 } //GEN-LAST: event_jButton5ActionPerformed
257
258 private void jTable1MouseClicked(java.awt.event.MouseEvent evt) { //
259     GEN-FIRST: event_jTable1MouseClicked
260     int columna = this.jTable1.getSelectedColumn();
261     int fila = this.jTable1.getSelectedRow();
262     this.jButton6.setEnabled(false);
263     ArrayList<String> term = new ArrayList<>();
264     term = buscarTerm();
265     int i=0;
266     int x=0;
267     while(i<term.size()){
268         if (this.tlr.getTAccion().getMatrizAccion().getValueAt(fila ,
269         0).toString().equals(term.get(i))){
270             JOptionPane.showConfirmDialog(null, "Esta fila no se
271             puede rellenar, debido a que el símbolo "+term.get
272             (i)+" aparece el primero en una de las producciones
273             .", "Salir",JOptionPane.CLOSED_OPTION);
274             x=-1;
275             break;
276         }else
277             i++;
278     }
279     if (x==0){
280         if (this.tlr.getTAccion().getMatrizAccion().getValueAt(fila ,
281         columna) == null){
282             String str = "E"+this.jComboBox1.getSelectedItem().
283             toString().substring(0, 1);
284             this.tlr.getTAccion().getMatrizAccion().setValueAt(str ,
285             fila , columna);
286             this.jTable1.setDefaultRenderer (Object.class , new
287             MiRender());
288         }
289         if (this.tlr.getTAccion().getMatrizAccion().getValueAt(fila ,
290         columna).toString().startsWith("E")){
291             this.jButton6.setEnabled(true);
292         }
293         this.gramatica.setTlr(this.tlr);
294     }
295 } //GEN-LAST: event_jTable1MouseClicked
296
297 private ArrayList<String> buscarTerm(){
298     ArrayList<String> term = new ArrayList<>();

```



```

329         break;
330     }else{
331         j++;
332     }
333 }
334 }
335 i++;
336 }
337 this.jButton7.setText("Eliminar Reducciones");
338 }else{ //red par: Eliminar reducciones
339
340     int i=0;
341     int j=1;
342
343     while(i < tabla.getMatrizAccion().getRowCount()){
344         j=1;
345         while(j < tabla.getMatrizAccion().getColumnCount()){
346             if(tabla.getMatrizAccion().getValueAt(i, j)!=null){
347                 if(tabla.getMatrizAccion().getValueAt(i, j).
348                     toString().contains("*")){
349                     tabla.getMatrizAccion().setValueAt(null, i,
350                         j);
351                 }
352             }
353             j++;
354         }
355         i++;
356     }
357     this.jButton7.setText("Completar con Reducciones");
358     this.tlr.setTAccion(tabla);
359     this.ventanaPadre.gramatica.setTlr(this.tlr);
360 }//GEN-LAST:event_jButton7ActionPerformed
361
362 void completarTabla(int fila, String reduccion){
363     int i=0;
364
365     while(i < this.tlr.getTAccion().getMatrizAccion().
366         getColumnCount()){
367         if(this.tlr.getTAccion().getMatrizAccion().getValueAt(fila,
368             i) == null){
369             this.tlr.getTAccion().getMatrizAccion().setValueAt(
370                 reduccion, fila, i);
371         }
372         i++;
373     }
374 }
375
376 // Variables declaration - do not modify//GEN-BEGIN:variables
377 private javax.swing.JButton jButton1;
378 private javax.swing.JButton jButton2;
379 private javax.swing.JButton jButton3;
380 private javax.swing.JButton jButton4;

```

```

377     private javax.swing.JButton jButton5;
378     private javax.swing.JButton jButton6;
379     private javax.swing.JButton jButton7;
380     private javax.swing.JComboBox jComboBox1;
381     private javax.swing.JLabel jLabel1;
382     private javax.swing.JLabel jLabel2;
383     private javax.swing.JScrollPane jScrollPane1;
384     private javax.swing.JTable jTable1;
385     // End of variables declaration//GEN-END:variables
386 }

```

### 2.5.12. PanelNuevaSimDescPaso1.java

```

1  //SimAS / Simulador
2  //Panel nueva simulacion descendente paso 1
3
4  package es.uco.simas.simulador;
5
6  import javax.swing.JOptionPane;
7  import es.uco.simas.util.gramatica.Gramatica;
8  import es.uco.simas.util.gramatica.Produccion;
9  import java.util.ArrayList;
10 import javax.swing.DefaultListModel;
11
12 /**
13  * @author Vanesa
14  */
15 public class PanelNuevaSimDescPaso1 extends javax.swing.JPanel {
16
17     private VentanaSimuladorDesc ventanaPadre;
18     private Gramatica gramatica;
19
20     public PanelNuevaSimDescPaso1(VentanaSimuladorDesc ventanaPadre) {
21         initComponents();
22         this.ventanaPadre = ventanaPadre;
23         this.gramatica = this.ventanaPadre.getGramatica();
24
25         int recursiva = this.recursividad(this.gramatica);
26         int factorizar = this.factorizar(this.gramatica);
27         if(recursiva == 1){
28             this.jLabel2.setText("La gramática original era recursiva
29 por la izquierda.");
30             this.jLabel2.setForeground(new java.awt.Color(162, 7, 7));
31         }
32         if(factorizar == 1){
33             this.jLabel3.setText("La gramática original no estaba
34 factorizada.");
35             this.jLabel3.setForeground(new java.awt.Color(162, 7, 7));
36         }
37         if(factorizar == 0 && recursiva == 0){

```

```

36         this.jLabel2.setText("La gramática original es correcta.")
37         ;
38         this.jLabel2.setForeground(new java.awt.Color(33, 77, 72));
39     }
40 }
41 /**
42  * This method is called from within the constructor to initialize
43  * the form.
44  * WARNING: Do NOT modify this code. The content of this method is
45  * always
46  * regenerated by the Form Editor.
47  */
48 @SuppressWarnings("unchecked")
49 // <editor-fold defaultstate="collapsed" desc="Generated Code">
50 GEN-BEGIN: initComponents
51 private void initComponents() {
52     JLabel1 = new javax.swing.JLabel();
53     JButton1 = new javax.swing.JButton();
54     JButton2 = new javax.swing.JButton();
55     JButton3 = new javax.swing.JButton();
56     JButton4 = new javax.swing.JButton();
57     JButton5 = new javax.swing.JButton();
58     JScrollPane1 = new javax.swing.JScrollPane();
59     JList1 = new javax.swing.JList();
60     JLabel2 = new javax.swing.JLabel();
61     JLabel3 = new javax.swing.JLabel();
62
63     setBackground(new java.awt.Color(233, 244, 244));
64
65     JLabel1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
66     JLabel1.setText("Producciones para la simulacion Descendente:");
67     ;
68
69     JButton1.setText("Cancelar");
70     JButton1.addActionListener(new java.awt.event.ActionListener()
71     {
72         public void actionPerformed(java.awt.event.ActionEvent evt)
73         {
74             JButton1ActionPerformed(evt);
75         }
76     });
77
78     JButton2.setIcon(new javax.swing.ImageIcon(getClass().
79         getResource("/es/uco/simas/resources/ultimo.png"))); //
80     NOI18N
81     JButton2.addActionListener(new java.awt.event.ActionListener()
82     {
83         public void actionPerformed(java.awt.event.ActionEvent evt)
84         {
85             JButton2ActionPerformed(evt);
86         }
87     });

```

```

78     });
79
80     jButton3.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/siguiente.png"))); //
        NOI18N
81     jButton3.addActionListener(new java.awt.event.ActionListener()
        {
82         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
83             jButton3ActionPerformed(evt);
84         }
85     });
86
87     jButton4.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/anterior.png"))); //
        NOI18N
88     jButton4.setEnabled(false);
89
90     jButton5.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/primer.png"))); //
        NOI18N
91     jButton5.setEnabled(false);
92
93     jScrollPane1.setViewportView(jList1);
94
95     jLabel2.setForeground(new java.awt.Color(44, 103, 71));
96
97     jLabel3.setForeground(new java.awt.Color(162, 7, 7));
98
99     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
        this);
100    this.setLayout(layout);
101    layout.setHorizontalGroup(
102        layout.createParallelGroup(javax.swing.GroupLayout.
            Alignment.LEADING)
103        .addGroup(layout.createSequentialGroup()
104            .addGap(62, 62, 62)
105            .addGroup(layout.createParallelGroup(javax.swing.
                GroupLayout.Alignment.LEADING)
106                .addGroup(layout.createSequentialGroup()
107                    .addGroup(layout.createParallelGroup(javax.
                        swing.GroupLayout.Alignment.LEADING)
108                        .addGroup(layout.createSequentialGroup()
109                            .addGap(26, 26, 26)
110                            .addComponent(jButton1)
111                            .addPreferredGap(javax.swing.
                                LayoutStyle.ComponentPlacement.
                                    RELATED, javax.swing.GroupLayout.
                                        DEFAULT_SIZE, Short.MAX_VALUE)
112                        .addComponent(jButton5)
113                        .addPreferredGap(javax.swing.
                            LayoutStyle.ComponentPlacement.
                                RELATED)

```

```

114         .addComponent(jButton4)
115         .addPreferredGap(javax.swing.
            LayoutStyle.ComponentPlacement.
            RELATED)
116         .addComponent(jButton3)
117         .addPreferredGap(javax.swing.
            LayoutStyle.ComponentPlacement.
            RELATED))
118         .addGroup(layout.createSequentialGroup())
119         .addComponent(jScrollPane1, javax.swing.
            GroupLayout.PREFERRED_SIZE, 400,
            javax.swing.GroupLayout.
            PREFERRED_SIZE)
120         .addPreferredGap(javax.swing.
            LayoutStyle.ComponentPlacement.
            RELATED, 23, Short.MAX_VALUE)))
121         .addComponent(jButton2))
122     .addGroup(layout.createSequentialGroup())
123     .addGap(10, 10, 10)
124     .addGroup(layout.createParallelGroup(javax.
        swing.GroupLayout.Alignment.LEADING)
125         .addComponent(jLabel1)
126         .addComponent(jLabel2))
127     .addGap(0, 0, Short.MAX_VALUE)))
128     .addContainerGap())
129     .addGroup(layout.createSequentialGroup())
130     .addGap(82, 82, 82)
131     .addComponent(jLabel3)
132     .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
        Short.MAX_VALUE))
133 );
134 layout.setVerticalGroup(
135     layout.createParallelGroup(javax.swing.GroupLayout.
        Alignment.LEADING)
136     .addGroup(layout.createSequentialGroup())
137     .addGap(20, 20, 20)
138     .addComponent(jLabel1)
139     .addPreferredGap(javax.swing.LayoutStyle.
        ComponentPlacement.RELATED)
140     .addComponent(jLabel2)
141     .addPreferredGap(javax.swing.LayoutStyle.
        ComponentPlacement.RELATED, 60, Short.MAX_VALUE)
142     .addComponent(jLabel3)
143     .addPreferredGap(javax.swing.LayoutStyle.
        ComponentPlacement.RELATED)
144     .addComponent(jScrollPane1, javax.swing.GroupLayout.
        PREFERRED_SIZE, 290, javax.swing.GroupLayout.
        PREFERRED_SIZE)
145     .addGap(26, 26, 26)
146     .addGroup(layout.createParallelGroup(javax.swing.
        GroupLayout.Alignment.BASELINE)
147         .addComponent(jButton1)
148         .addComponent(jButton2)

```



```

149         .addComponent(jButton3)
150         .addComponent(jButton4)
151         .addComponent(jButton5))
152         .addGap(20, 20, 20))
153     );
154 }// </editor-fold>//GEN-END: initComponents
155
156 public int recursividad(Gramatica gramatica){
157     int recursiva = 0;
158     DefaultListModel noTerminales = gramatica.getNoTerminales();
159     DefaultListModel produc = gramatica.getProducciones();
160     DefaultListModel produc2 = new DefaultListModel();
161     StringBuilder nombre = new StringBuilder();
162     StringBuilder prod = new StringBuilder();
163     DefaultListModel eliminar = new DefaultListModel();
164     ArrayList<Produccion> pr = new ArrayList();
165     int i=0;
166     int l=0;
167     int elim = -1;
168
169     while(i < produc.size()){
170         String valor = produc.elementAt(i).toString();
171         String antec = "";
172         String [] separado;
173         separado = valor.split(" ");
174         antec = separado[0];
175
176         produc2.addElement(produc.elementAt(i));
177
178         if(antec.equals(separado[2])){
179             recursiva = 1;
180             produc2.removeElement(produc.elementAt(i));
181             //Se crea el símbolo no terminal "Nombre"
182             nombre.delete(0,nombre.length());
183             nombre = nombre.append(antec);
184             nombre = nombre.append(" ");
185             int k=0;
186             int iguales = 0;
187             while(k < noTerminales.size()){
188                 if(noTerminales.elementAt(k).toString().equals(
189                     nombre.toString())){
190                     iguales = 1;
191                     break;
192                 }else
193                     k++;
194             }
195             if(iguales == 0){
196                 noTerminales.addElement(nombre.toString());
197                 this.gramatica.setNoTerminales(noTerminales);
198             }
199             prod.delete(0,prod.length());
200             k=0;

```

```

201         while(k < produc.size()){
202             String [] aux=produc.elementAt(k).toString().
                split(" ");
203
204             if(aux[0].equals(antec) && !aux[0].toString().
                equals(aux[2].toString()) && produc.
                getElementAt(k) != produc.getElementAt(i) && i
                != k){
205                 eliminar.addElement(produc.elementAt(k));
206
207                 String valor2 = produc.elementAt(k).toString
                    ();
208                 String [] separado2 = valor2.split(" ");
209                 prod.delete(0,prod.length());
210                 prod = prod.append(antec).append(" \u2192");
211
212                 l=2;
213                 while(l<separado2.length){
214                     if(!separado2[2].toString().equals("\u03b5"
                        ))
215                         prod = prod.append(" ").append(
                            separado2[1]);
216                     l++;
217                 }
218                 prod = prod.append(" ").append(nombre.toString
                    ());
219
220                 l=0;
221                 iguales = 0;
222                 while(l < produc2.size()){
223                     if(produc2.elementAt(l).toString().
                        equals(prod.toString())){
224                         iguales = 1;
225                         break;
226                     }else
227                         l++;
228                 }
229                 if(iguales == 0)
230                     produc2.addElement(prod.toString());
231
232             }
233             k++;
234         }
235
236         k=0;
237         while(k < produc2.size()){
238             String [] aux=produc2.elementAt(k).toString().
                split(" ");
239             if(!aux[0].toString().equals(nombre) && !separado
                [3].toString().equals(aux[2].toString())){
240                 prod.delete(0,prod.length());
241                 prod = prod.append(nombre.toString());
242                 prod = prod.append(" \u2192");

```

```

243         int j=3;
244         while(j<separado.length){
245             prod = prod.append(" ");
246             prod = prod.append(separado[j]);
247             j++;
248         }
249         prod = prod.append(" ").append(nombre.toString
250             ());
251         l=0;
252         iguales = 0;
253         while(l < produc2.size()){
254             if(produc2.elementAt(l).toString().
255                 equals(prod.toString())){
256                 iguales = 1;
257                 break;
258             }else
259                 l++;
260         }
261         if(iguales == 0)
262             produc2.addElement(prod.toString());
263         break;
264     }else
265         k++;
266 }
267
268     prod.delete(0,prod.length());
269     prod = prod.append(nombre.toString()).append("
270         \u2192 ").append("\u03b5");
271
272     l=0;
273     iguales = 0;
274     while(l < produc2.size()){
275         if(produc2.elementAt(l).toString().
276             equals(prod.toString())){
277             iguales = 1;
278             break;
279         }else
280             l++;
281     }
282     if(iguales == 0)
283         produc2.addElement(prod.toString());
284
285     }else
286         elim = -1;
287     i++;
288 }
289
290 l=0;
291
292 while(l < eliminar.size()){
293     int j=0;
294     while(j < produc2.size()){

```

```

291         if (produc2.getElementAt(j).toString().equals(eliminar.
292             getElementAt(l).toString())){
293             produc2.remove(j);
294         }
295         j++;
296     }
297     l++;
298 }
299
300 this.gramatica.setProducciones(new DefaultListModel());
301 this.gramatica.setProducciones(produc2);
302 this.jList1.setModel(produc2);
303 this.gramatica = gramatica;
304 this.ventanaPadre.setGramatica(gramatica);
305 return recursiva;
306 }
307
308 public int factorizar(Gramatica gramatica){
309     int factorizar = 0;
310     ArrayList<Integer> comprobado = new ArrayList<>();
311     DefaultListModel noTerminales = gramatica.getNoTerminales();
312     DefaultListModel produc = gramatica.getProducciones();
313     DefaultListModel produc2 = new DefaultListModel();
314     DefaultListModel aux;
315     StringBuilder nombre = new StringBuilder();
316     StringBuilder prod = new StringBuilder();
317     int i=0;
318     int j=0;
319     int k=0;
320     int iguales=0;
321
322     this.ventanaPadre.setPr(this.gramatica.getPr());
323     this.ventanaPadre.setNoTerminales(this.gramatica.getNoTerm());
324
325     while(i < produc.size()){
326         String valor = produc.getElementAt(i).toString();
327         String [] separado = valor.split(" ");
328         String antec = separado[0];
329         aux = new DefaultListModel();
330         k=0;
331
332         if(!comprobado.isEmpty()){
333             j=0;
334             while(j < comprobado.size()){
335                 if(comprobado.get(j) == i){
336                     i++;
337                     if(i<produc.size()){
338                         valor = produc.getElementAt(i).toString();
339                         separado = valor.split(" ");
340                         antec = separado[0];
341                     }
342                     break;
343                 }
344             }
345         }
346     }

```

```

343         j++;
344     }
345 }
346
347 j=0;
348 while(j < produc.size()){
349     String valor2 = produc.elementAt(j).toString();
350     String [] separado2 = valor2.split(" ");
351     String antec2 = separado2[0];
352
353     if(antec.equals(antec2) && separado[2].toString().
        equals(separado2[2].toString()) && i != j){
354         comprobado.add(j);
355         factorizar = 1;
356         aux.addElement(produc.elementAt(j));
357     }
358     j++;
359 }
360
361 if(!aux.isEmpty()){
362     j=0;
363     while(j<aux.size()){
364         String valor2 = aux.elementAt(j).toString();
365         String antec2 = "";
366         String [] separado2 = valor2.split(" ");
367         antec2 = separado2[0];
368         nombre.delete(0,nombre.length());
369         nombre = nombre.append(antec);
370         nombre = nombre.append(" ");
371         k=0;
372         iguales = 0;
373         while(k < noTerminales.size()){
374             if(noTerminales.elementAt(k).toString().
                equals(nombre.toString())){
375                 iguales = 1;
376                 break;
377             }else
378                 k++;
379         }
380         if(iguales == 0){
381             noTerminales.addElement(nombre.toString());
382         }else{
383
384             k=0;
385             while(k < produc2.size()){
386                 String valor3 = produc2.elementAt(k).
                    toString();
387                 String [] separado3 = valor3.split(" ");
388                 if(antec.equals(separado3[0]) && !separado
                    [2].equals(separado3[2])){
389                     nombre = nombre.append(" ");
390                     int m=0;
391                     iguales = 0;

```

```

392         while(m < noTerminales.size()){
393             if(noTerminales.elementAt(m).
                toString().equals(nombre.
                toString())){
394                 iguales = 1;
395                 break;
396             }else
397                 m++;
398         }
399         if(iguales == 0){
400             noTerminales.addElement(nombre.
                toString());
401         }
402         break;
403     }else
404         k++;
405 }
406
407 }
408
409 prod.delete(0,prod.length());
410 prod = prod.append(antec).append(" \u2192 ").append
    (separado[2]).append(" ").append(nombre);
411 k=0;
412 iguales=0;
413 while(k < produc2.size()){
414     if(produc2.elementAt(k).toString().equals(
        prod.toString())){
415         iguales = 1;
416         break;
417     }else
418         k++;
419 }
420 if(iguales == 0){
421     produc2.addElement(prod.toString());
422 }
423
424 prod.delete(0,prod.length());
425 prod = prod.append(nombre).append(" \u2192");
426
427 if(separado.length > 3 ){
428     k=3;
429     while(k < separado.length){
430         prod = prod.append(" ").append(separado[k])
            ;
431         k++;
432     }
433 }else{
434     prod = prod.append(" \u03b5");
435 }
436 k=0;
437 iguales=0;
438 while(k < produc2.size()){

```

```

439         if(produc2.getElementAt(k).toString().equals(
440             prod.toString())){
441             iguales = 1;
442             break;
443         }else
444             k++;
445     }
446     if(iguales == 0){
447         produc2.addElement(prod.toString());
448     }
449     prod.delete(0,prod.length());
450     prod = prod.append(nombre).append(" \u2192");
451
452     if(separado2.length > 3){
453         k=3;
454         while(k < separado2.length){
455             prod = prod.append(" ").append(separado2[k
456                 ]);
457             k++;
458         }
459     }else{
460         prod = prod.append(" \u03b5");
461     }
462     k=0;
463     iguales=0;
464     while(k < produc2.size()){
465         if(produc2.getElementAt(k).toString().equals(
466             prod.toString())){
467             iguales = 1;
468             break;
469         }else
470             k++;
471     }
472     if(iguales == 0){
473         produc2.addElement(prod.toString());
474     }
475     j++;
476 }else
477     produc2.addElement(produc.get(i));
478
479     i++;
480 }
481
482 if(produc2.size() != 0){
483     this.gramatica.setProducciones(produc2);
484     this.gramatica.setNoTerminales(noTerminales);
485     this.ventanaPadre.setGramatica(gramatica);
486     this.jList1.setModel(produc2);
487     this.gramatica = gramatica;
488 }

```

```

489         return factorizar;
490     }
491
492     private void jButton1ActionPerformed(java.awt.event.ActionEvent evt
493     ) {//GEN-FIRST:event_jButton1ActionPerformed
494         int conf = JOptionPane.showConfirmDialog(null, "¿Desea salir
495         del asistente de la simulacion de la gramática?", "Salir",
496         JOptionPane.YES_NO_OPTION);
497
498         if (conf==0)
499             this.ventanaPadre.dispose();
500     }//GEN-LAST:event_jButton1ActionPerformed
501
502     private void jButton2ActionPerformed(java.awt.event.ActionEvent evt
503     ) {//GEN-FIRST:event_jButton2ActionPerformed
504         PanelNuevaSimDescPaso2 paso2 = new PanelNuevaSimDescPaso2(this.
505         ventanaPadre);
506         this.gramatica.generarConjPrim();
507         this.gramatica.generarConjSig();
508         paso2.construirConjuntos(this.gramatica);
509
510         PanelNuevaSimDescPaso3 paso3 = new PanelNuevaSimDescPaso3(this.
511         ventanaPadre);
512         paso3.ConstruirTPredictiva(this.gramatica);
513
514         this.ventanaPadre.cambiarPaso(4);
515     }//GEN-LAST:event_jButton2ActionPerformed
516
517     private void jButton3ActionPerformed(java.awt.event.ActionEvent evt
518     ) {//GEN-FIRST:event_jButton3ActionPerformed
519         this.ventanaPadre.cambiarPaso(2);
520     }//GEN-LAST:event_jButton3ActionPerformed
521
522     // Variables declaration - do not modify//GEN-BEGIN:variables
523     private javax.swing.JButton jButton1;
524     private javax.swing.JButton jButton2;
525     private javax.swing.JButton jButton3;
526     private javax.swing.JButton jButton4;
527     private javax.swing.JButton jButton5;
528     private javax.swing.JLabel jLabel1;
529     private javax.swing.JLabel jLabel2;
530     private javax.swing.JLabel jLabel3;
531     private javax.swing.JList jList1;
532     private javax.swing.JScrollPane jScrollPane1;
533     // End of variables declaration//GEN-END:variables
534 }

```

### 2.5.13. PanelNuevaSimDescPaso2.java



```

1 //SimAS / Simulador
2 //Panel nueva simulacion descendente paso 2
3
4 package es.uco.simas.simulador;
5
6 import es.uco.simas.util.gramatica.Gramatica;
7 import es.uco.simas.util.gramatica.NoTerminal;
8 import es.uco.simas.util.gramatica.Terminal;
9 import java.util.ArrayList;
10 import javax.swing.JOptionPane;
11 import javax.swing.JTable;
12 import javax.swing.table.DefaultTableModel;
13
14 /**
15  * @author Vanesa
16  */
17 public class PanelNuevaSimDescPaso2 extends javax.swing.JPanel {
18
19     private VentanaSimuladorDesc ventanaPadre;
20     private DefaultTableModel modeloConjuntos ;
21     private JTable tblConjuntos ;
22
23     public PanelNuevaSimDescPaso2(VentanaSimuladorDesc ventanaPadre) {
24         DefaultTableModel tabla = new DefaultTableModel();
25         this.modeloConjuntos =tabla;
26         initComponents();
27
28         this.ventanaPadre =ventanaPadre;
29
30         this.jTable1.getColumnModel().getColumn(0).setPreferredWidth
31             (90);
32         this.jTable1.getColumnModel().getColumn(1).setPreferredWidth
33             (250);
34         this.jTable1.getColumnModel().getColumn(2).setPreferredWidth
35             (290);
36         this.jTable1.setEnabled( false );
37     }
38
39     public void construirConjuntos(Gramatica gramatica) {
40
41         ArrayList<NoTerminal> noTerminales= null;
42         ArrayList<Terminal> conjuntoPrimero= null;
43         ArrayList<Terminal> conjuntoSiguiente= null;
44
45         NoTerminal noTerminal= null;
46         String strConjuntoPrimero= null;
47         String strConjuntoSiguiente= null;
48
49         int i= 0;
50         int j=0;
51
52         this.modeloConjuntos =new DefaultTableModel();

```

```

50     this.modeloConjuntos.addColumn(" Simbolo");
51     this.modeloConjuntos.addColumn(" Conjunto  primero");
52     this.modeloConjuntos.addColumn(" Conjunto  siguiente");
53     this.jTable1.setModel( this.modeloConjuntos);
54     this.jTable1.getColumnModel().getColumn(0).setPreferredWidth
        (95);
55     this.jTable1.getColumnModel().getColumn(1).setPreferredWidth
        (250);
56     this.jTable1.getColumnModel().getColumn(2).setPreferredWidth
        (290);
57     this.jTable1.setEnabled( false);
58
59     noTerminales = gramatica.getNoTerm();
60
61     i=0;
62     while(i < noTerminales.size()){
63         strConjuntoPrimero = "";
64         strConjuntoSiguiente = "";
65         noTerminal = noTerminales.get(i);
66         conjuntoPrimero = noTerminal.getPrimeros();
67         conjuntoSiguiente = noTerminal.getSiguientes();
68
69         j=0;
70         while(j < conjuntoPrimero.size()){
71             StringBuilder cadena = new StringBuilder();
72             strConjuntoPrimero = cadena.append(strConjuntoPrimero).
                append(" ").append(conjuntoPrimero.get(j).getNombre
                ()).toString();
73
74             j++;
75         }
76
77         j=0;
78         while(j < conjuntoSiguiente.size()){
79             StringBuilder cadena = new StringBuilder();
80             strConjuntoSiguiente = cadena.append(
                strConjuntoSiguiente).append(" ").append(
                conjuntoSiguiente.get(j).getNombre()).toString();
81
82             j++;
83         }
84
85         Object [] linea = new Object[] {
86             noTerminal.getValor(), strConjuntoPrimero,
                strConjuntoSiguiente
87         };
88         this.modeloConjuntos.addRow(linea);
89
90         i++;
91     }
92 }
93
94 /**

```

```

95      * This method is called from within the constructor to initialize
96      * the form.
97      * WARNING: Do NOT modify this code. The content of this method is
98      * always
99      * regenerated by the Form Editor.
100      */
101      @SuppressWarnings("unchecked")
102      // <editor-fold defaultstate="collapsed" desc="Generated Code">
103      GEN-BEGIN: initComponents
104      private void initComponents() {
105
106          jScrollPane1 = new javax.swing.JScrollPane();
107          jTable1 = new javax.swing.JTable();
108          jLabel1 = new javax.swing.JLabel();
109          jButton1 = new javax.swing.JButton();
110          jButton2 = new javax.swing.JButton();
111          jButton3 = new javax.swing.JButton();
112          jButton4 = new javax.swing.JButton();
113          jButton5 = new javax.swing.JButton();
114
115          setBackground(new java.awt.Color(233, 244, 244));
116          setMinimumSize(new java.awt.Dimension(45, 80));
117
118          jTable1.setModel(new javax.swing.table.DefaultTableModel(
119              new Object [][] {
120                  {null, null, null},
121                  {null, null, null},
122                  {null, null, null},
123                  {null, null, null}
124              },
125              new String [] {
126                  "Símbolo", "Conjunto Primero", "Conjunto Siguiete"
127              }
128          ) {
129              boolean[] canEdit = new boolean [] {
130                  false, false, false
131              };
132
133              public boolean isCellEditable(int rowIndex, int columnIndex) {
134                  return canEdit [columnIndex];
135              }
136          });
137          jScrollPane1.setViewportView(jTable1);
138
139          jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
140          jLabel1.setText("Conjunto Primero y Siguiete");
141
142          jButton1.setText("Cancelar");
143          jButton1.addActionListener(new java.awt.event.ActionListener()
144              {
145              public void actionPerformed(java.awt.event.ActionEvent evt)
146              {

```

```

142         jButton1ActionPerformed(evt);
143     }
144 });
145
146     jButton2.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/ultimo.png"))); //
        NOI18N
147     jButton2.addActionListener(new java.awt.event.ActionListener()
        {
148         public void actionPerformed(java.awt.event.ActionEvent evt)
        {
149             jButton2ActionPerformed(evt);
150         }
151     });
152
153     jButton3.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/siguiente.png"))); //
        NOI18N
154     jButton3.addActionListener(new java.awt.event.ActionListener()
        {
155         public void actionPerformed(java.awt.event.ActionEvent evt)
        {
156             jButton3ActionPerformed(evt);
157         }
158     });
159
160     jButton4.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/anterior.png"))); //
        NOI18N
161     jButton4.addActionListener(new java.awt.event.ActionListener()
        {
162         public void actionPerformed(java.awt.event.ActionEvent evt)
        {
163             jButton4ActionPerformed(evt);
164         }
165     });
166
167     jButton5.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/primer.png"))); //
        NOI18N
168     jButton5.addActionListener(new java.awt.event.ActionListener()
        {
169         public void actionPerformed(java.awt.event.ActionEvent evt)
        {
170             jButton5ActionPerformed(evt);
171         }
172     });
173
174     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
        this);
175     this.setLayout(layout);
176     layout.setHorizontalGroup(

```

```

177         layout.createParallelGroup(javax.swing.GroupLayout.
178             Alignment.LEADING)
179         .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
180             layout.createSequentialGroup())
181             .addGap(24, 24, 24)
182             .addGroup(layout.createParallelGroup(javax.swing.
183                 GroupLayout.Alignment.TRAILING)
184                 .addComponent(jScrollPane1, javax.swing.GroupLayout.
185                     DEFAULT_SIZE, 556, Short.MAX_VALUE)
186                 .addGroup(layout.createSequentialGroup())
187                     .addComponent(jButton1)
188                     .addPreferredGap(javax.swing.LayoutStyle.
189                         ComponentPlacement.RELATED, javax.swing.
190                             GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
191                     .addComponent(jButton5)
192                     .addPreferredGap(javax.swing.LayoutStyle.
193                         ComponentPlacement.UNRELATED)
194                     .addComponent(jButton4)
195                     .addPreferredGap(javax.swing.LayoutStyle.
196                         ComponentPlacement.RELATED)
197                     .addComponent(jButton3)
198                     .addPreferredGap(javax.swing.LayoutStyle.
199                         ComponentPlacement.RELATED)
200                     .addComponent(jButton2)))
201             .addGap(33, 33, 33))
202         .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
203             layout.createSequentialGroup())
204             .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
205                 Short.MAX_VALUE)
206             .addComponent(jLabel1)
207             .addGap(156, 156, 156))
208     );
209     layout.setVerticalGroup(
210         layout.createParallelGroup(javax.swing.GroupLayout.
211             Alignment.LEADING)
212         .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
213             layout.createSequentialGroup())
214             .addContainerGap()
215             .addComponent(jLabel1)
216             .addGap(43, 43, 43)
217             .addComponent(jScrollPane1, javax.swing.GroupLayout.
218                 PREFERRED_SIZE, 313, javax.swing.GroupLayout.
219                     PREFERRED_SIZE)
220             .addPreferredGap(javax.swing.LayoutStyle.
221                 ComponentPlacement.RELATED, 40, Short.MAX_VALUE)
222             .addGroup(layout.createParallelGroup(javax.swing.
223                 GroupLayout.Alignment.LEADING)
224                 .addGroup(javax.swing.GroupLayout.Alignment.
225                     TRAILING, layout.createParallelGroup(javax.
226                         swing.GroupLayout.Alignment.LEADING)
227                         .addComponent(jButton2, javax.swing.GroupLayout.
228                             Alignment.TRAILING)
229                         .addComponent(jButton1))

```

```

210         .addComponent(jButton3, javax.swing.GroupLayout.
211             Alignment.TRAILING)
212         .addComponent(jButton4, javax.swing.GroupLayout.
213             Alignment.TRAILING)
213         .addComponent(jButton5, javax.swing.GroupLayout.
214             Alignment.TRAILING))
214         .addContainerGap();
215 }// </editor-fold>//GEN-END: initComponents
216
217 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt
218 ) { //GEN-FIRST: event_jButton1ActionPerformed
219     int conf = JOptionPane.showConfirmDialog(null, "¿Desea salir
220     del asistente de la simulacion de la gramática?", "Salir",
221     JOptionPane.YES_NO_OPTION);
222
223     if (conf==0)
224         this.ventanaPadre.dispose();
225 }//GEN-LAST: event_jButton1ActionPerformed
226
227 private void jButton5ActionPerformed(java.awt.event.ActionEvent evt
228 ) { //GEN-FIRST: event_jButton5ActionPerformed
229     this.ventanaPadre.cambiarPaso(1);
230 }//GEN-LAST: event_jButton5ActionPerformed
231
232 private void jButton4ActionPerformed(java.awt.event.ActionEvent evt
233 ) { //GEN-FIRST: event_jButton4ActionPerformed
234     this.ventanaPadre.cambiarPaso(1);
235 }//GEN-LAST: event_jButton4ActionPerformed
236
237 private void jButton3ActionPerformed(java.awt.event.ActionEvent evt
238 ) { //GEN-FIRST: event_jButton3ActionPerformed
239     this.ventanaPadre.cambiarPaso(3);
240 }//GEN-LAST: event_jButton3ActionPerformed
241
242 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt
243 ) { //GEN-FIRST: event_jButton2ActionPerformed
244     PanelNuevaSimDescPaso3 paso3 = new PanelNuevaSimDescPaso3(this.
245     ventanaPadre);
246     paso3.ConstruirTPredictiva(this.ventanaPadre.gramatica);
247
248     this.ventanaPadre.cambiarPaso(4);
249 }//GEN-LAST: event_jButton2ActionPerformed
250
251 // Variables declaration - do not modify//GEN-BEGIN: variables
252 private javax.swing.JButton jButton1;
253 private javax.swing.JButton jButton2;
254 private javax.swing.JButton jButton3;
255 private javax.swing.JButton jButton4;
256 private javax.swing.JButton jButton5;
257 private javax.swing.JLabel jLabel1;
258 private javax.swing.JScrollPane jScrollPane1;

```

```

252     private javax.swing.JTable jTable1;
253     // End of variables declaration//GEN-END:variables
254 }

```

### 2.5.14. PanelNuevaSimDescPaso3.java

```

1  //SimAS / Simulador
2  //Panel nueva simulacion descendente paso 3
3
4  package es.uco.simas.simulador;
5
6  import es.uco.simas.editor.TablaPredictiva;
7  import es.uco.simas.util.gramatica.*;
8  import javax.swing.JOptionPane;
9  import javax.swing.table.DefaultTableModel;
10
11 /**
12  * @author Vanesa
13  */
14 public class PanelNuevaSimDescPaso3 extends javax.swing.JPanel {
15
16     private VentanaSimuladorDesc ventanaPadre;
17     private Gramatica gramatica;
18     private DefaultTableModel modeloConjuntos ;
19
20     public PanelNuevaSimDescPaso3(VentanaSimuladorDesc ventanaPadre) {
21         DefaultTableModel tabla = new DefaultTableModel();
22         this.modeloConjuntos = tabla;
23         initComponents();
24         this.ventanaPadre = ventanaPadre;
25         this.gramatica = ventanaPadre.getGramatica();
26     }
27
28     public void ConstruirTPredictiva(Gramatica gramatica){
29
30         this.gramatica.generarTPredictiva();
31         TablaPredictiva tpredictiva = this.gramatica.getTPredictiva();
32
33         this.jTable1.setModel(tpredictiva.getTabla());
34     }
35
36     /**
37      * This method is called from within the constructor to initialize
38      * the form.
39      * WARNING: Do NOT modify this code. The content of this method is
40      * always
41      * regenerated by the Form Editor.
42      */
43     @SuppressWarnings("unchecked")

```

```

42 // <editor-fold defaultstate="collapsed" desc="Generated Code">//
    GEN-BEGIN: initComponents
43 private void initComponents() {
44
45     jLabel1 = new javax.swing.JLabel();
46     jButton1 = new javax.swing.JButton();
47     jButton2 = new javax.swing.JButton();
48     jButton4 = new javax.swing.JButton();
49     jButton5 = new javax.swing.JButton();
50     jScrollPane1 = new javax.swing.JScrollPane();
51     jTable1 = new javax.swing.JTable();
52     jButton6 = new javax.swing.JButton();
53
54     setBackground(new java.awt.Color(233, 244, 244));
55
56     jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
57     jLabel1.setText("Tabla Predictiva");
58
59     jButton1.setText("Cancelar");
60     jButton1.addActionListener(new java.awt.event.ActionListener()
61     {
62         public void actionPerformed(java.awt.event.ActionEvent evt)
63         {
64             jButton1ActionPerformed(evt);
65         }
66     });
67     jButton2.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
68     jButton2.setIcon(new javax.swing.ImageIcon(getClass().
69         getResource("/es/uco/simas/resources/ultimo.png"))); //
        NOI18N
70     jButton2.addActionListener(new java.awt.event.ActionListener()
71     {
72         public void actionPerformed(java.awt.event.ActionEvent evt)
73         {
74             jButton2ActionPerformed(evt);
75         }
76     });
77     jButton4.setIcon(new javax.swing.ImageIcon(getClass().
78         getResource("/es/uco/simas/resources/anterior.png"))); //
        NOI18N
79     jButton4.addActionListener(new java.awt.event.ActionListener()
80     {
81         public void actionPerformed(java.awt.event.ActionEvent evt)
82         {
83             jButton4ActionPerformed(evt);
84         }
85     });
86     jButton5.setIcon(new javax.swing.ImageIcon(getClass().
87         getResource("/es/uco/simas/resources/primero.png"))); //
        NOI18N

```



```

82     jButton5.addActionListener(new java.awt.event.ActionListener()
83     {
84         public void actionPerformed(java.awt.event.ActionEvent evt)
85         {
86             jButton5ActionPerformed(evt);
87         }
88     });
89     jTable1.setModel(new javax.swing.table.DefaultTableModel(
90         new Object [][] {
91             {null, null, null, null},
92             {null, null, null, null},
93             {null, null, null, null},
94             {null, null, null, null}
95         },
96         new String [] {
97             "", "", "", ""
98         }
99     ) {
100         boolean[] canEdit = new boolean [] {
101             false, false, false, false
102         };
103         public boolean isCellEditable(int rowIndex, int columnIndex
104         ) {
105             return canEdit [columnIndex];
106         }
107     });
108     jScrollPane1.setViewportViewView(jTable1);
109     jButton6.setIcon(new javax.swing.ImageIcon(getClass().
110         getResource("/es/uco/simas/resources/siguiente.png"))); //
111     NOI18N
112     jButton6.addActionListener(new java.awt.event.ActionListener()
113     {
114         public void actionPerformed(java.awt.event.ActionEvent evt)
115         {
116             jButton6ActionPerformed(evt);
117         }
118     });
119     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
120         this);
121     this.setLayout(layout);
122     layout.setHorizontalGroup(
123         layout.createParallelGroup(javax.swing.GroupLayout.

```

```

124         .addComponent(jScrollPane1 , javax.swing.
           GroupLayout.PREFERRED_SIZE, 652, javax.
           swing.GroupLayout.PREFERRED_SIZE)
125         .addContainerGap(39, Short.MAX_VALUE))
126     .addGroup(layout.createSequentialGroup())
127     .addComponent(jButton1)
128     .addPreferredGap(javax.swing.LayoutStyle.
           ComponentPlacement.RELATED, javax.swing.
           GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
129     .addComponent(jButton5)
130     .addPreferredGap(javax.swing.LayoutStyle.
           ComponentPlacement.RELATED)
131     .addComponent(jButton4)
132     .addPreferredGap(javax.swing.LayoutStyle.
           ComponentPlacement.RELATED)
133     .addComponent(jButton6)
134     .addPreferredGap(javax.swing.LayoutStyle.
           ComponentPlacement.RELATED)
135     .addComponent(jButton2)
136     .addGap(36, 36, 36)))
137     .addGroup(layout.createSequentialGroup())
138     .addGap(262, 262, 262)
139     .addComponent(jLabel1)
140     .addGap(0, 0, Short.MAX_VALUE))
141 );
142 layout.setVerticalGroup(
143     layout.createParallelGroup(javax.swing.GroupLayout.
           Alignment.LEADING)
144     .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
           layout.createSequentialGroup()
145         .addContainerGap()
146         .addComponent(jLabel1)
147         .addGap(28, 28, 28)
148         .addComponent(jScrollPane1 , javax.swing.GroupLayout.
           DEFAULT_SIZE, 400, Short.MAX_VALUE)
149         .addGap(18, 18, 18)
150         .addGroup(layout.createParallelGroup(javax.swing.
           GroupLayout.Alignment.LEADING)
151             .addGroup(layout.createParallelGroup(javax.swing.
           GroupLayout.Alignment.LEADING)
152                 .addGroup(javax.swing.GroupLayout.Alignment.
           TRAILING, layout.createSequentialGroup()
153                     .addComponent(jButton1)
154                     .addGap(5, 5, 5))
155                 .addGroup(javax.swing.GroupLayout.Alignment.
           TRAILING, layout.createParallelGroup(javax.
           swing.GroupLayout.Alignment.LEADING)
156                     .addComponent(jButton2 , javax.swing.
           GroupLayout.PREFERRED_SIZE, 31, javax.
           swing.GroupLayout.PREFERRED_SIZE)
157                     .addComponent(jButton6)
158                     .addComponent(jButton4)))
159         .addGroup(layout.createSequentialGroup())

```

```

160         .addGap(2, 2, 2)
161         .addComponent(jButton5)))
162     .addContainerGap()
163 );
164 }// </editor-fold>//GEN-END: initComponents
165
166 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt
167 ) { //GEN-FIRST: event_jButton1ActionPerformed
168     int conf = JOptionPane.showConfirmDialog(null, "¿Desea salir
169     del asistente de la simulacion de la gramática?", "Salir",
170     JOptionPane.YES_NO_OPTION);
171
172     if (conf==0)
173         this.ventanaPadre.dispose();
174 }//GEN-LAST: event_jButton1ActionPerformed
175
176 private void jButton5ActionPerformed(java.awt.event.ActionEvent evt
177 ) { //GEN-FIRST: event_jButton5ActionPerformed
178     this.ventanaPadre.cambiarPaso(1);
179 }//GEN-LAST: event_jButton5ActionPerformed
180
181 private void jButton4ActionPerformed(java.awt.event.ActionEvent evt
182 ) { //GEN-FIRST: event_jButton4ActionPerformed
183     this.ventanaPadre.cambiarPaso(3);
184 }//GEN-LAST: event_jButton4ActionPerformed
185
186 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt
187 ) { //GEN-FIRST: event_jButton2ActionPerformed
188     this.ventanaPadre.cambiarPaso(4);
189 }//GEN-LAST: event_jButton2ActionPerformed
190
191 private void jButton6ActionPerformed(java.awt.event.ActionEvent evt
192 ) { //GEN-FIRST: event_jButton6ActionPerformed
193     this.ventanaPadre.cambiarPaso(4);
194 }//GEN-LAST: event_jButton6ActionPerformed
195
196 // Variables declaration - do not modify//GEN-BEGIN: variables
197 private javax.swing.JButton jButton1;
198 private javax.swing.JButton jButton2;
199 private javax.swing.JButton jButton4;
200 private javax.swing.JButton jButton5;
201 private javax.swing.JButton jButton6;
202 private javax.swing.JLabel jLabel1;
203 private javax.swing.JScrollPane jScrollPane1;
204 private javax.swing.JTable jTable1;
205 // End of variables declaration//GEN-END: variables
206 }

```

### 2.5.15. PanelNuevaSimDescPaso4.java

```

1 //SimAS / Simulador
2 //Panel nueva simulacion descendente paso 4
3
4 package es.uco.simas.simulador;
5
6 import es.uco.simas.editor.FuncionError;
7 import es.uco.simas.util.gramatica.Gramatica;
8 import java.util.ArrayList;
9 import javax.swing.DefaultListModel;
10 import javax.swing.JOptionPane;
11
12 /**
13  * @author vanesa
14  */
15 public class PanelNuevaSimDescPaso4 extends javax.swing.JPanel {
16
17     Gramatica gramatica;
18     VentanaSimuladorDesc ventanaPadre;
19     private int funError=0;
20
21     public PanelNuevaSimDescPaso4(VentanaSimuladorDesc ventanaPadre) {
22         initComponents();
23         this.ventanaPadre = ventanaPadre;
24         this.gramatica = ventanaPadre.getGramatica();
25
26         if(this.gramatica.getTPredictiva().getFunError().size() != 0){
27             funcionError();
28             this.funError = 1;
29         }
30         if(funError ==0){
31             this.jButtonFinalizar.setVisible(true);
32             this.jButtonUltimo.setVisible(false);
33             this.jButtonSiguiente.setEnabled(false);
34             this.jButtonEliminar.setEnabled(false);
35         }
36     }
37
38     public void funcionError(){
39         DefaultListModel lista = new DefaultListModel();
40         ArrayList<FuncionError> funError = this.gramatica.
41             getTPredictiva().getFunError();
42         StringBuilder string = new StringBuilder();
43         int i=0;
44
45         while(i < funError.size()){
46             string = new StringBuilder();
47             int accion;
48             string = string.append(funError.get(i).getIdentificador());
49             string = string.append(" - ");
50             accion = funError.get(i).getAccion();
51             if(accion == 1)

```

```

51         string.append("Insertar un SÃmbolo en la Entrada: ");
52     if(accion == 2)
53         string.append("Borrar un SÃmbolo de la Entrada");
54     if(accion == 3)
55         string.append("Modificar un SÃmbolo de la Entrada: ");
56     if(accion == 4)
57         string.append("Insertar un SÃmbolo de la Pila: ");
58     if(accion == 5)
59         string.append("Borrar un SÃmbolo de la Pila");
60     if(accion == 6)
61         string.append("Modificar un SÃmbolo de la Pila: ");
62     if(accion == 7)
63         string.append("Terminar el anÃlisis");
64     if(accion == 1 || accion ==3 || accion ==4 || accion ==6)
65         string.append(funError.get(i).getSimbolo().getNombre())
66         ;
67     lista.add(i, string);
68     i++;
69
70 }
71 this.jList1.setModel(lista);
72
73 if(this.jList1.getModel().getSize() !=0){
74     this.jButtonFinalizar.setVisible(false);
75
76     this.jButtonSiguiente.setEnabled(true);
77     this.jButtonNueva.setEnabled(true);
78     this.jButtonEliminar.setEnabled(true);
79 }else{
80     this.jButtonNueva.setEnabled(true);
81     this.jButtonEliminar.setEnabled(false);
82     this.jButtonFinalizar.setVisible(true);
83
84     this.jButtonSiguiente.setEnabled(false);
85 }
86
87 }
88
89 /**
90  * This method is called from within the constructor to initialize
91  * the form.
92  * WARNING: Do NOT modify this code. The content of this method is
93  * always
94  * regenerated by the Form Editor.
95  */
96 @SuppressWarnings("unchecked")
97 // <editor-fold defaultstate="collapsed" desc="Generated Code">
98 GEN-BEGIN: initComponents
99 private void initComponents() {
100
101     jLabel1 = new javax.swing.JLabel();
102     jButtonCancelar = new javax.swing.JButton();

```

```

100     jButtonUltimo = new javax.swing.JButton();
101     jButtonSiguiente = new javax.swing.JButton();
102     jButton3 = new javax.swing.JButton();
103     jButton4 = new javax.swing.JButton();
104     jScrollPane1 = new javax.swing.JScrollPane();
105     jList1 = new javax.swing.JList();
106     jCheckBox1 = new javax.swing.JCheckBox();
107     jButtonNueva = new javax.swing.JButton();
108     jButtonEliminar = new javax.swing.JButton();
109     jButtonFinalizar = new javax.swing.JButton();
110
111     setBackground(new java.awt.Color(233, 244, 244));
112
113     jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
114     jLabel1.setText("Funciones de Error");
115
116     jButtonCancelar.setText("Cancelar");
117     jButtonCancelar.addActionListener(new java.awt.event.
        ActionListener() {
118         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
119             jButtonCancelarActionPerformed(evt);
120         }
121     });
122
123     jButtonUltimo.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/ultimo.png"))); //
        NOI18N
124     jButtonUltimo.addActionListener(new java.awt.event.
        ActionListener() {
125         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
126             jButtonUltimoActionPerformed(evt);
127         }
128     });
129
130     jButtonSiguiente.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/siguiente.png"))); //
        NOI18N
131     jButtonSiguiente.addActionListener(new java.awt.event.
        ActionListener() {
132         public void actionPerformed(java.awt.event.ActionEvent evt)
            {
133             jButtonSiguienteActionPerformed(evt);
134         }
135     });
136
137     jButton3.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/anterior.png"))); //
        NOI18N
138     jButton3.addActionListener(new java.awt.event.ActionListener()
        {

```

```

139         public void actionPerformed(java.awt.event.ActionEvent evt)
140         {
141             jButton3ActionPerformed(evt);
142         }
143     });
144     jButton4.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/primero.png"))); //
        NOI18N
145     jButton4.addActionListener(new java.awt.event.ActionListener()
        {
146         public void actionPerformed(java.awt.event.ActionEvent evt)
147         {
148             jButton4ActionPerformed(evt);
149         }
150     });
151     jScrollPane1.setViewportView(list1);
152
153     checkBox1.setText("No utilizar Funciones de error");
154     checkBox1.addActionListener(new java.awt.event.ActionListener
        () {
155         public void actionPerformed(java.awt.event.ActionEvent evt)
156         {
157             checkBox1ActionPerformed(evt);
158         }
159     });
160     jButtonNueva.setText("Nueva Funcion Error ");
161     jButtonNueva.addActionListener(new java.awt.event.
        ActionListener() {
162         public void actionPerformed(java.awt.event.ActionEvent evt)
163         {
164             jButtonNuevaActionPerformed(evt);
165         }
166     });
167     jButtonEliminar.setText("Eliminar Funcion Error");
168     jButtonEliminar.addActionListener(new java.awt.event.
        ActionListener() {
169         public void actionPerformed(java.awt.event.ActionEvent evt)
170         {
171             jButtonEliminarActionPerformed(evt);
172         }
173     });
174     jButtonFinalizar.setFont(new java.awt.Font("Ubuntu", 1, 15));
        // NOI18N
175     jButtonFinalizar.setText("Finalizar");
176     jButtonFinalizar.addActionListener(new java.awt.event.
        ActionListener() {
177         public void actionPerformed(java.awt.event.ActionEvent evt)
        {

```





```

221         .addComponent(jLabel1)
222         .addPreferredGap(javax.swing.LayoutStyle.
           ComponentPlacement.RELATED)
223         .addGroup(layout.createParallelGroup(javax.swing.
           GroupLayout.Alignment.LEADING)
224             .addGroup(layout.createSequentialGroup())
225                 .addGap(45, 45, 45)
226                 .addComponent(jButtonNueva)
227                 .addGap(58, 58, 58)
228                 .addComponent(jButtonEliminar)
229                 .addGap(0, 146, Short.MAX_VALUE))
230             .addComponent(jScrollPane1))
231         .addPreferredGap(javax.swing.LayoutStyle.
           ComponentPlacement.UNRELATED)
232         .addComponent(jCheckBox1)
233         .addGap(48, 48, 48)
234         .addGroup(layout.createParallelGroup(javax.swing.
           GroupLayout.Alignment.LEADING)
235             .addComponent(jButtonFinalizar)
236             .addComponent(jButtonUltimo)
237             .addComponent(jButton3)
238             .addComponent(jButtonCancelar)
239             .addComponent(jButton4)
240             .addComponent(jButtonSiguiente))
241         .addGap(19, 19, 19)
242     );
243 }// </editor-fold>//GEN-END: initComponents
244
245 private void jButtonNuevaActionPerformed(java.awt.event.ActionEvent
           evt) { //GEN-FIRST: event_jButtonNuevaActionPerformed
246     NuevaFuncionError error = new NuevaFuncionError(this.gramatica,
           this);
247     error.setLocationRelativeTo(null);
248     error.setVisible(true);
249 }//GEN-LAST: event_jButtonNuevaActionPerformed
250
251 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt
           ) { //GEN-FIRST: event_jButton1ActionPerformed
252     int conf = JOptionPane.showConfirmDialog(null, "¿Desea salir
           del asistente de la simulacion de la gramática?", "Salir",
           JOptionPane.YES_NO_OPTION);
253
254     if (conf==0)
255         this.ventanaPadre.dispose();
256 }//GEN-LAST: event_jButton1ActionPerformed
257
258 private void jCheckBox1ActionPerformed(java.awt.event.ActionEvent
           evt) { //GEN-FIRST: event_jCheckBox1ActionPerformed
259     if (this.jCheckBox1.isSelected()) {
260         this.jButtonNueva.setEnabled(false);
261         this.jButtonEliminar.setEnabled(false);
262         this.jButtonFinalizar.setVisible(true);
263         this.jButtonUltimo.setVisible(false);

```

```

264         this.jButtonSiguiente.setEnabled(false);
265     }else{
266         if(this.jList1.getModel().getSize()!=0){
267             this.jButtonFinalizar.setVisible(false);
268             this.jButtonUltimo.setVisible(true);
269             this.jButtonSiguiente.setEnabled(true);
270             this.jButtonNueva.setEnabled(true);
271             this.jButtonEliminar.setEnabled(true);
272         }else{
273             this.jButtonNueva.setEnabled(true);
274             this.jButtonEliminar.setEnabled(false);
275             this.jButtonFinalizar.setVisible(true);
276             this.jButtonUltimo.setVisible(false);
277             this.jButtonSiguiente.setEnabled(false);
278         }
279     }
280 }
281 }//GEN-LAST:event_jCheckBox1ActionPerformed
282
283 private void jButtonEliminarActionPerformed(java.awt.event.
    ActionEvent evt) { //GEN-FIRST:
    event_jButtonEliminarActionPerformed
284     DefaultListModel modelo= null;
285     int seleccion= this.jList1.getSelectedIndex();
286     String funcion = this.jList1.getModel().getElementAt(seleccion)
        .toString();
287     ArrayList<FuncionError> funError = this.gramatica.
        getTPredictiva().getFunError();
288     String id = funcion.substring(0, 1);
289     int num = Integer.parseInt(id);
290     int i=0;
291
292     if(seleccion != -1) {
293         modelo=(DefaultListModel) this.jList1.getModel();
294         modelo.remove(seleccion);
295
296         while(i < funError.size()){
297             if(funError.get(i).getIdentificador() == num){
298                 funError.remove(i);
299             }else
300                 i++;
301         }
302         this.gramatica.getTPredictiva().setFunError(funError);
303         this.ventanaPadre.setGramatica(this.gramatica);
304     }
305
306     if(this.jList1.getModel().getSize()!=0){
307         this.jButtonFinalizar.setVisible(false);
308         this.jButtonUltimo.setVisible(true);
309         this.jButtonSiguiente.setEnabled(true);
310         this.jButtonNueva.setEnabled(true);
311         this.jButtonEliminar.setEnabled(true);
312     }else{

```

```

313         this.jButtonNueva.setEnabled(true);
314         this.jButtonEliminar.setEnabled(false);
315         this.jButtonFinalizar.setVisible(true);
316         this.jButtonUltimo.setVisible(false);
317         this.jButtonSiguiente.setEnabled(false);
318     }
319     } //GEN-LAST: event_jButtonEliminarActionPerformed
320
321     private void jButtonSiguienteActionPerformed(java.awt.event.
        ActionEvent evt) { //GEN-FIRST:
        event_jButtonSiguienteActionPerformed
322         this.ventanaPadre.cambiarPaso(5);
323     } //GEN-LAST: event_jButtonSiguienteActionPerformed
324
325     private void jButton2ActionPerformed(java.awt.event.ActionEvent evt
        ) { //GEN-FIRST: event_jButton2ActionPerformed
326         this.ventanaPadre.cambiarPaso(3);
327     } //GEN-LAST: event_jButton2ActionPerformed
328
329     private void jButton3ActionPerformed(java.awt.event.ActionEvent evt
        ) { //GEN-FIRST: event_jButton3ActionPerformed
330         this.ventanaPadre.cambiarPaso(1);
331     } //GEN-LAST: event_jButton3ActionPerformed
332
333     private void jButtonFinalizarActionPerformed(java.awt.event.
        ActionEvent evt) { //GEN-FIRST:
        event_jButtonFinalizarActionPerformed
334         this.ventanaPadre.finalizarAsistente();
335     } //GEN-LAST: event_jButtonFinalizarActionPerformed
336
337     private void jButtonUltimoActionPerformed(java.awt.event.
        ActionEvent evt) { //GEN-FIRST:
        event_jButtonUltimoActionPerformed
338         this.ventanaPadre.cambiarPaso(5);
339     } //GEN-LAST: event_jButtonUltimoActionPerformed
340
341     private void jButtonCancelarActionPerformed(java.awt.event.
        ActionEvent evt) { //GEN-FIRST:
        event_jButtonCancelarActionPerformed
342         int conf = JOptionPane.showConfirmDialog(null, "¿Desea salir
        del asistente de la simulacion de la gramática?", "Salir"
        ,JOptionPane.YES_NO_OPTION);
343
344         if (conf==0)
345             this.ventanaPadre.dispose();
346     } //GEN-LAST: event_jButtonCancelarActionPerformed
347
348     private void jButton4ActionPerformed(java.awt.event.ActionEvent evt
        ) { //GEN-FIRST: event_jButton4ActionPerformed
349         this.ventanaPadre.cambiarPaso(1);
350     } //GEN-LAST: event_jButton4ActionPerformed
351
352

```

```

353 // Variables declaration – do not modify//GEN-BEGIN:variables
354 private javax.swing.JButton jButton3;
355 private javax.swing.JButton jButton4;
356 private javax.swing.JButton jButtonCancelar;
357 private javax.swing.JButton jButtonEliminar;
358 private javax.swing.JButton jButtonFinalizar;
359 private javax.swing.JButton jButtonNueva;
360 private javax.swing.JButton jButtonSiguiente;
361 private javax.swing.JButton jButtonUltimo;
362 private javax.swing.JCheckBox jCheckBox1;
363 private javax.swing.JLabel jLabel1;
364 private javax.swing.JList jList1;
365 private javax.swing.JScrollPane jScrollPane1;
366 // End of variables declaration//GEN-END:variables
367 }

```

### 2.5.16. PanelNuevaSimDescPaso5.java

```

1 //SimAS / Simulador
2 //Panel nueva simulacion descendente paso 5
3
4 package es.uco.simas.simulador;
5
6 import es.uco.simas.editor.FuncionError;
7 import es.uco.simas.editor.TablaPredictiva;
8 import es.uco.simas.util.gramatica.Gramatica;
9 import es.uco.simas.util.gramatica.Produccion;
10 import es.uco.simas.util.gramatica.Terminal;
11 import java.util.ArrayList;
12 import javax.swing.DefaultComboBoxModel;
13 import javax.swing.JOptionPane;
14 import javax.swing.table.DefaultTableModel;
15
16 /**
17  * @author Vanesa
18  */
19 public class PanelNuevaSimDescPaso5 extends javax.swing.JPanel {
20
21     private VentanaSimuladorDesc ventanaPadre;
22     private Gramatica gramatica;
23     private DefaultTableModel modeloConjuntos ;
24     private TablaPredictiva tpredictiva;
25     int red = 0;
26
27     public PanelNuevaSimDescPaso5(VentanaSimuladorDesc ventanaPadre) {
28         DefaultTableModel tabla = new DefaultTableModel();
29         this.modeloConjuntos = tabla;
30         initComponents();
31         this.ventanaPadre = ventanaPadre;
32         this.gramatica = ventanaPadre.getGramatica();

```

```

33     this.jButton6.setEnabled(false);
34     this.tpredictiva = this.gramatica.getTPredictiva();
35     this.jTable1.setModel(this.tpredictiva.getTabla());
36     this.inicializarCombos(this.tpredictiva);
37     this.anadirTerminales();
38     this.jTable1.setDefaultRenderer (Object.class, new MiRender());
39 }
40
41 private void anadirTerminales(){
42     ArrayList<Terminal> term = this.ventanaPadre.getGramatica().
43         getTerm();
44     DefaultTableModel tabla = this.tpredictiva.getTabla();
45     int fila = this.tpredictiva.getTabla().getRowCount();
46     int columna = 1;
47     int i=0;
48     while(i < term.size()){
49         Object [] linea = new Object [] {
50             term.get(i).getNombre()
51         };
52         tabla.addRow(linea);
53
54         i++;
55     }
56     Object [] linea = new Object [] {
57         "$"
58     };
59     tabla.addRow(linea);
60
61     this.tpredictiva.setTabla(tabla);
62     i = fila;
63     while(i < tabla.getRowCount()-1){
64         this.tpredictiva.setCeldaPredictiva("<HTML> <FONT COLOR=\"
65             blue\">Emparejar</FONT></HTML>", i, columna);
66         i++;
67         columna++;
68     }
69     this.tpredictiva.setCeldaPredictiva("<HTML> <FONT COLOR=\"blue
70         \">Aceptar</FONT></HTML>", i, columna);
71 }
72
73 /**
74  * This method is called from within the constructor to initialize
75  * the form.
76  * WARNING: Do NOT modify this code. The content of this method is
77  * always
78  * regenerated by the Form Editor.
79  */
80 // @SuppressWarnings("unchecked")
81 // <editor-fold defaultstate="collapsed" desc="Generated Code">
82 // GEN-BEGIN: initComponents
83 private void initComponents() {

```

```

80      jLabel1 = new javax.swing.JLabel();
81      jButton1 = new javax.swing.JButton();
82      jButton2 = new javax.swing.JButton();
83      jButton3 = new javax.swing.JButton();
84      jButton4 = new javax.swing.JButton();
85      jScrollPane1 = new javax.swing.JScrollPane();
86      jTable1 = new javax.swing.JTable();
87      jButton5 = new javax.swing.JButton();
88      jComboBox1 = new javax.swing.JComboBox();
89      jLabel2 = new javax.swing.JLabel();
90      jButton6 = new javax.swing.JButton();
91      jButton7 = new javax.swing.JButton();
92
93      setBackground(new java.awt.Color(233, 244, 244));
94
95      jLabel1.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
96      jLabel1.setText("Incluir Funciones de Error");
97
98      jButton1.setFont(new java.awt.Font("Tahoma", 1, 15)); // NOI18N
99      jButton1.setText("Finalizar");
100     jButton1.addActionListener(new java.awt.event.ActionListener()
101     {
102         public void actionPerformed(java.awt.event.ActionEvent evt)
103         {
104             jButton1ActionPerformed(evt);
105         }
106     });
107
108     jButton2.setIcon(new javax.swing.ImageIcon(getClass().
109         getResource("/es/uco/simas/resources/siguiente.png"))); //
110         NOI18N
111     jButton2.setEnabled(false);
112
113     jButton3.setIcon(new javax.swing.ImageIcon(getClass().
114         getResource("/es/uco/simas/resources/anterior.png"))); //
115         NOI18N
116     jButton3.addActionListener(new java.awt.event.ActionListener()
117     {
118         public void actionPerformed(java.awt.event.ActionEvent evt)
119         {
120             jButton3ActionPerformed(evt);
121         }
122     });
123
124     jButton4.setIcon(new javax.swing.ImageIcon(getClass().
125         getResource("/es/uco/simas/resources/primero.png"))); //
126         NOI18N
127     jButton4.addActionListener(new java.awt.event.ActionListener()
128     {
129         public void actionPerformed(java.awt.event.ActionEvent evt)
130         {
131             jButton4ActionPerformed(evt);
132         }
133     });

```

```

121     });
122
123     jTable1.setModel(new javax.swing.table.DefaultTableModel(
124         new Object [][] {
125             {null, null, null, null},
126             {null, null, null, null},
127             {null, null, null, null},
128             {null, null, null, null}
129         },
130         new String [] {
131             "", "", "", ""
132         }
133     ) {
134         boolean[] canEdit = new boolean [] {
135             false, false, false, false
136         };
137
138         public boolean isCellEditable(int rowIndex, int columnIndex
139             ) {
140             return canEdit [columnIndex];
141         }
142     });
143     jTable1.addMouseListener(new java.awt.event.MouseAdapter() {
144         public void mouseClicked(java.awt.event.MouseEvent evt) {
145             jTable1MouseClicked(evt);
146         }
147     });
148     jScrollPane1.setViewportViewView(jTable1);
149
150     jButton5.setFont(new java.awt.Font("Tahoma", 1, 15)); // NOI18N
151     jButton5.setText("Cancelar");
152     jButton5.addActionListener(new java.awt.event.ActionListener()
153     {
154         public void actionPerformed(java.awt.event.ActionEvent evt)
155         {
156             jButton5ActionPerformed(evt);
157         }
158     });
159
160     jLabel2.setText("Pulse en la tabla para insertar la funcion de
161         error:");
162
163     jButton6.setText("Eliminar Funcion de Error");
164     jButton6.addActionListener(new java.awt.event.ActionListener()
165     {
166         public void actionPerformed(java.awt.event.ActionEvent evt)
167         {
168             jButton6ActionPerformed(evt);
169         }
170     });
171
172     jButton7.setText("Rellenar con producciones  $\tilde{A}@\epsilon$ ");

```

```

167     jButton7.addActionListener(new java.awt.event.ActionListener()
168     {
169         public void actionPerformed(java.awt.event.ActionEvent evt)
170         {
171             jButton7ActionPerformed(evt);
172         }
173     });
174
175     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
176     this);
177     this.setLayout(layout);
178     layout.setHorizontalGroup(
179     layout.createParallelGroup(javax.swing.GroupLayout.
180     Alignment.LEADING)
181     .addGroup(layout.createSequentialGroup()
182     .addGroup(layout.createParallelGroup(javax.swing.
183     GroupLayout.Alignment.LEADING)
184     .addGroup(layout.createSequentialGroup()
185     .addGap(26, 26, 26)
186     .addGroup(layout.createParallelGroup(javax.
187     swing.GroupLayout.Alignment.LEADING)
188     .addComponent(jScrollPane1, javax.swing.
189     GroupLayout.PREFERRED_SIZE, 582, javax.
190     swing.GroupLayout.PREFERRED_SIZE)
191     .addGroup(layout.createSequentialGroup()
192     .addComponent(jButton5)
193     .addPreferredGap(javax.swing.
194    .LayoutStyle.ComponentPlacement.
195     RELATED, 309, Short.MAX_VALUE)
196     .addComponent(jButton4)
197     .addPreferredGap(javax.swing.
198    .LayoutStyle.ComponentPlacement.
199     RELATED)
200     .addComponent(jButton3)
201     .addPreferredGap(javax.swing.
202    .LayoutStyle.ComponentPlacement.
203     RELATED)
204     .addComponent(jButton2)
205     .addPreferredGap(javax.swing.
206    .LayoutStyle.ComponentPlacement.
207     RELATED)
208     .addComponent(jButton1))))
209     .addGroup(layout.createSequentialGroup()
210     .addGap(172, 172, 172)
211     .addComponent(jLabel1))
212     .addGroup(layout.createSequentialGroup()
213     .addGap(121, 121, 121)
214     .addGroup(layout.createParallelGroup(javax.
215     swing.GroupLayout.Alignment.LEADING)
216     .addComponent(jLabel2)
217     .addGroup(layout.createSequentialGroup()
218     .addGap(118, 118, 118)

```



```

202         .addComponent(jComboBox1, javax.swing.
        GroupLayout.PREFERRED_SIZE, javax.
        swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.
        PREFERRED_SIZE) ) ) )
203     .addContainerGap()
204     .addGroup(layout.createSequentialGroup())
205     .addGap(102, 102, 102)
206     .addComponent(jButton6)
207     .addGap(31, 31, 31)
208     .addComponent(jButton7)
209     .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
        Short.MAX_VALUE) )
210 );
211 layout.setVerticalGroup(
212     layout.createParallelGroup(javax.swing.GroupLayout.
        Alignment.LEADING)
213     .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
        layout.createSequentialGroup()
214         .addGap(15, 15, 15)
215         .addComponent(jLabel1)
216         .addGap(12, 12, 12)
217         .addComponent(jLabel2)
218         .addGap(18, 18, 18)
219         .addComponent(jComboBox1, javax.swing.GroupLayout.
            PREFERRED_SIZE, javax.swing.GroupLayout.
            DEFAULT_SIZE, javax.swing.GroupLayout.
            PREFERRED_SIZE)
220         .addGap(18, 18, 18)
221         .addComponent(jScrollPane1, javax.swing.GroupLayout.
            DEFAULT_SIZE, 351, Short.MAX_VALUE)
222         .addGap(32, 32, 32)
223         .addGroup(layout.createParallelGroup(javax.swing.
            GroupLayout.Alignment.BASELINE)
224             .addComponent(jButton6)
225             .addComponent(jButton7))
226         .addGap(18, 18, 18)
227         .addGroup(layout.createParallelGroup(javax.swing.
            GroupLayout.Alignment.LEADING)
228             .addGroup(layout.createParallelGroup(javax.swing.
                GroupLayout.Alignment.TRAILING)
229                 .addComponent(jButton2)
230                 .addComponent(jButton1)
231                 .addComponent(jButton3)
232                 .addComponent(jButton4))
233             .addComponent(jButton5))
234         .addGap(25, 25, 25))
235 );
236 }// </editor-fold> //GEN-END: initComponents
237
238 private void inicializarCombos(TablaPredictiva tpredictiva) {
239     ArrayList<String> listaCombo = new ArrayList();
240     ArrayList<FuncionError> funError = tpredictiva.getFunError();

```

```

241     StringBuilder string = new StringBuilder();
242     int i= 0;
243
244     while(i < funError.size()) {
245         string = new StringBuilder();
246         int accion;
247         string = string.append(funError.get(i).getIdentificador());
248         string = string.append(" - ");
249         accion = funError.get(i).getAccion();
250         if(accion == 1)
251             string.append("Insertar un S mbolo en la Entrada: ");
252         if(accion == 2)
253             string.append("Borrar un S mbolo de la Entrada");
254         if(accion == 3)
255             string.append("Modificar un S mbolo de la Entrada: ");
256         if(accion == 4)
257             string.append("Insertar un S mbolo de la Pila: ");
258         if(accion == 5)
259             string.append("Borrar un S mbolo de la Pila");
260         if(accion == 6)
261             string.append("Modificar un S mbolo de la Pila: ");
262         if(accion == 7)
263             string.append("Terminar el an lisis");
264         if(accion == 1 || accion == 3 || accion == 4 || accion == 6)
265             string.append(funError.get(i).getSimbolo().getNombre());
266
267         ;
268
269         listaCombo.add(string.toString());
270         i++;
271     }
272     DefaultComboBoxModel combo = new DefaultComboBoxModel(
273         listaCombo.toArray());
274     this.jComboBox1.setModel(combo);
275 }
276
277 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt
278 ) { //GEN-FIRST:event_jButton1ActionPerformed
279     this.gramatica.setTPredictiva(this.tpredictiva);
280     this.ventanaPadre.finalizarAsistente();
281     this.ventanaPadre.dispose();
282 } //GEN-LAST:event_jButton1ActionPerformed
283
284 private void jButton5ActionPerformed(java.awt.event.ActionEvent evt
285 ) { //GEN-FIRST:event_jButton5ActionPerformed
286     int conf = JOptionPane.showConfirmDialog(null, "  Desea salir
del asistente de la simulacion de la gram tica?", "Salir",
JOptionPane.YES_NO_OPTION);
287
288     if(conf==0)
289         this.ventanaPadre.dispose();
290 } //GEN-LAST:event_jButton5ActionPerformed

```

```

287 private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {//
    GEN-FIRST: event_jTable1MouseClicked
288     int columna = this.jTable1.getSelectedColumn();
289     int fila = this.jTable1.getSelectedRow();
290     this.jButton6.setEnabled(false);
291     ArrayList<String> term = new ArrayList<>();
292     term = buscarTerm();
293     int i=0;
294     int x=0;
295     while(i<term.size()){
296         if(this.tpredictiva.getCeldaPredictiva(fila, 0).toString().
            equals(term.get(i))){
297             JOptionPane.showConfirmDialog(null, "Esta fila no se
                puede rellenar, debido a que el símbolo \""+term.
                get(i)+"\" no aparece en la parte derecha de una
                produccion en un lugar que no sea el primero.", "
                Salir",JOptionPane.CLOSED_OPTION);
298             //Esta fila no se puede rellenar, debido a que el
                símbolo "+term.get(i)+" aparece el primero en una
                de las producciones.", "Salir",JOptionPane.
                CLOSED_OPTION);
299             x=-1;
300             break;
301         }else
302             i++;
303     }
304     if(x==0){
305         if(this.tpredictiva.getCeldaPredictiva(fila, columna) ==
            null){
306             String str = "E"+this.jComboBox1.getSelectedItem().
                toString().substring(0, 1);
307             this.tpredictiva.setCeldaPredictiva(str, fila, columna)
                ;
308             // this.jTable1.setDefaultRenderer (Object.class, new
                MiRender());
309         }
310         if(this.tpredictiva.getCeldaPredictiva(fila, columna).
            toString().startsWith("E")){
311             this.jButton6.setEnabled(true);
312         }
313         this.gramatica.setTPredictiva(this.tpredictiva);
314     }
315 }//GEN-LAST: event_jTable1MouseClicked
316
317 private ArrayList<String> buscarTerm(){
318     ArrayList<String> term = new ArrayList<>();
319     int i=0;
320     while(i< this.gramatica.getPr().size()){
321         if(this.gramatica.isTerminal(this.gramatica.getPr().get(i).
            getConsec().get(0).getNombre()))
322             term.add(this.gramatica.getPr().get(i).getConsec().get
                (0).getNombre());
323         i++;

```

```

324     }
325     return term;
326 }
327
328 private void jButton3ActionPerformed(java.awt.event.ActionEvent evt
329 ) {//GEN-FIRST:event_jButton3ActionPerformed
330     this.ventanaPadre.cambiarPaso(4);
331 }//GEN-LAST:event_jButton3ActionPerformed
332
333 private void jButton4ActionPerformed(java.awt.event.ActionEvent evt
334 ) {//GEN-FIRST:event_jButton4ActionPerformed
335     this.ventanaPadre.cambiarPaso(1);
336 }//GEN-LAST:event_jButton4ActionPerformed
337
338 private void jButton6ActionPerformed(java.awt.event.ActionEvent evt
339 ) {//GEN-FIRST:event_jButton6ActionPerformed
340     int columna = this.jTable1.getSelectedColumn();
341     int fila = this.jTable1.getSelectedRow();
342     this.tpredictiva.setCeldaPredictiva(null, fila, columna);
343     this.jButton6.setEnabled(false);
344 }//GEN-LAST:event_jButton6ActionPerformed
345
346 private void jButton7ActionPerformed(java.awt.event.ActionEvent evt
347 ) {//GEN-FIRST:event_jButton7ActionPerformed
348     ArrayList<Produccion> prod = this.gramatica.getPr();
349     int i=0;
350     red++;
351
352     if(red %2 != 0){ //red es impar: completar con reducciones
353         while(i < prod.size()){
354             if(prod.get(i).getConsec().get(0).getNombre().equals("\
355 u03b5")){
356                 String ant = prod.get(i).getAntec().getSimboloNT().
357                 getNombre();
358                 int j=0;
359                 while(j < this.jTable1.getRowCount()){
360                     if(this.jTable1.getValueAt(j, 0).toString().
361                     equals(ant)){
362                         int k=1;
363                         while(k < this.jTable1.getColumnCount()){
364                             if(this.jTable1.getValueAt(j, k)==null)
365                             {
366                                 this.jTable1.setValueAt(i+1+"*", j
367                                     , k);
368                             }
369                             k++;
370                         }
371                     }
372                     j++;
373                 }
374             }
375             i++;
376         }
377     }

```

```

368         }
369         this.jButton7.setText("Eliminar producciones  $\tilde{A}@\epsilon$ ");
370     }else{ //red par: Eliminar reducciones
371
372         i=0;
373         int j=1;
374
375         while(i < this.jTable1.getRowCount()){
376             j=1;
377             while(j < this.jTable1.getColumnCount()){
378                 if(this.jTable1.getValueAt(i, j)!=null){
379                     if(this.jTable1.getValueAt(i, j).toString().
                        contains("*")){
380                         this.jTable1.setValueAt(null, i, j);
381                     }
382                 }
383                 j++;
384             }
385             i++;
386         }
387         this.jButton7.setText("Rellenar con producciones  $\tilde{A}@\epsilon$ ")
            ;
388     }
389     this.gramatica.setTPredictiva(this.tpredictiva);
390
391     }//GEN-LAST:event_jButton7ActionPerformed
392
393     // Variables declaration - do not modify//GEN-BEGIN:variables
394     private javax.swing.JButton jButton1;
395     private javax.swing.JButton jButton2;
396     private javax.swing.JButton jButton3;
397     private javax.swing.JButton jButton4;
398     private javax.swing.JButton jButton5;
399     private javax.swing.JButton jButton6;
400     private javax.swing.JButton jButton7;
401     private javax.swing.JComboBox jComboBox1;
402     private javax.swing.JLabel jLabel1;
403     private javax.swing.JLabel jLabel2;
404     private javax.swing.JScrollPane jScrollPane1;
405     private javax.swing.JTable jTable1;
406     // End of variables declaration//GEN-END:variables
407 }

```

### 2.5.17. Simulador.java

```

1 //SimAS / Simulador
2 //Simulador
3
4 package es.uco.simas.simulador;
5

```

```

6 import com.itextpdf.text.BadElementException;
7 import com.itextpdf.text.BaseColor;
8 import com.itextpdf.text.Chunk;
9 import com.itextpdf.text.DocumentException;
10 import com.itextpdf.text.Font;
11 import com.itextpdf.text.Image;
12 import com.itextpdf.text.PageSize;
13 import com.itextpdf.text.Paragraph;
14 import com.itextpdf.text.pdf.BaseFont;
15 import com.itextpdf.text.pdf.PdfPCell;
16 import com.itextpdf.text.pdf.PdfPTable;
17 import com.itextpdf.text.pdf.PdfWriter;
18 import com.itextpdf.text.pdf.draw.LineSeparator;
19 import es.uco.simas.centroayuda.AcercaDe;
20 import es.uco.simas.editor.Editor;
21 import es.uco.simas.editor.FuncionError;
22 import es.uco.simas.editor.TablaLR;
23 import es.uco.simas.editor.TablaPredictiva;
24 import es.uco.simas.util.gramatica.*;
25 import java.io.File;
26 import java.io.FileNotFoundException;
27 import java.io.FileOutputStream;
28 import java.io.IOException;
29 import java.net.MalformedURLException;
30 import java.util.ArrayList;
31 import java.util.logging.Level;
32 import java.util.logging.Logger;
33 import javax.swing.DefaultListModel;
34 import javax.swing.JFileChooser;
35 import javax.swing.JOptionPane;
36 import javax.swing.filechooser.FileNameExtensionFilter;
37 import javax.swing.table.DefaultTableModel;
38
39 /**
40  * @author vanesa
41  */
42 public class Simulador extends javax.swing.JFrame {
43
44     ArrayList<Terminal> cadenaEntrada = new ArrayList<>();
45     int metodoSimulacion;
46     int modoFuncionamiento;
47     int metodoAscendente = -1;
48     Gramatica gramatica;
49
50     public Simulador(int i, Gramatica gramatica, int met) {
51
52         this.setCadenaEntrada(null);
53         this.metodoSimulacion = i;
54         this.gramatica = gramatica;
55         this.metodoAscendente = met;
56
57         DefaultListModel model = this.gramatica.getProducciones();
58         DefaultListModel model2 = new DefaultListModel();

```

```

59      initComponents();
60      this.jTable3.setVisible(false);
61      this.jTable2.removeAll();
62      this.jTable3.removeAll();
63      this.jTable2.setDefaultRenderer (Object.class, new MiRender());
64      this.jTable3.setDefaultRenderer (Object.class, new MiRender());
65
66      if (this.metodoSimulacion == 2){
67          if (this.metodoAscendente == 0){
68              this.jLabelSimulador.setText("Simulacion Ascendente SLR
69              ");
69          }
70          if (this.metodoAscendente == 1){
71              this.jLabelSimulador.setText("Simulacion Ascendente LR-
72              Canonico");
72          }
73          if (this.metodoAscendente == 2){
74              this.jLabelSimulador.setText("Simulacion Ascendente
75              LALR");
75          }
76
77          this.jTable3.setVisible(true);
78          this.jLabelTabla.setText("Tabla LR");
79          this.jLabelAccion.setVisible(true);
80          this.jLabelIra.setVisible(true);
81          TablaLR tlr = this.gramatica.getTlr();
82          this.jTable2.setModel(tlr.getTAccion().getMatrizAccion());
83          this.jTable3.setModel(tlr.getTIrA().getMatrizIrA());
84          this.funError(tlr.getTAccion().getFunError());
85      }
86      if (this.metodoSimulacion == 1){
87          this.jTable3.setVisible(false);
88          this.jLabelSimulador.setText("Simulacion Descendente");
89          this.jLabelTabla.setText("Tabla Predictiva");
90          this.jLabelAccion.setVisible(false);
91          this.jLabelIra.setVisible(false);
92          TablaPredictiva tpred = this.gramatica.getTPredictiva();
93          this.jTable2.setModel(tpred.getTabla());
94          this.funError(tpred.getFunError());
95      }
96
97      if (this.gramatica.getProducciones() != null){
98          int j = 0;
99          Object obj;
100          obj = "P {";
101          model2.addElement(obj);
102          while (j < model.size()){
103              obj = "      "+(j+1)+"      "+model.getElementAt(j);
104              model2.addElement(obj);
105              j++;
106          }
107          obj = "}";
108          model2.addElement(obj);

```

```

109         this.jList1.setModel(model2);
110     }
111 }
112
113 void funError(ArrayList<FuncionError> funError){
114     StringBuilder string = new StringBuilder();
115     DefaultListModel lista = new DefaultListModel();
116     int j= 0;
117
118     while(j < funError.size()) {
119         string = new StringBuilder();
120         int accion;
121
122         string = string.append(funError.get(j).getIdentificador
123             ());
124         string = string.append(" - ");
125         accion = funError.get(j).getAccion();
126         if(accion == 1)
127             string.append("Insertar un S mbolo en la Entrada:
128                 ");
129         if(accion == 2)
130             string.append("Borrar un S mbolo de la Entrada");
131         if(accion == 3)
132             string.append("Modificar un S mbolo de la Entrada:
133                 ");
134         if(accion == 4)
135             string.append("Insertar un S mbolo de la Pila: ");
136         if(accion == 5)
137             string.append("Borrar un S mbolo de la Pila");
138         if(accion == 6)
139             string.append("Modificar un S mbolo de la Pila: ");
140         ;
141         if(accion == 7)
142             string.append("Terminar el an lisis");
143         if(accion == 1 || accion ==3 || accion ==4 || accion
144             ==6)
145             string.append(funError.get(j).getSimbolo().
146                 getNombre());
147
148         lista.addElement(string);
149         j++;
150     }
151     this.jList2.setModel(lista);
152 }
153
154 public Boolean generarInforme(String fichero) throws
155     DocumentException{
156     try {
157         String font = "fonts/arialuni.ttf";
158         com.itextpdf.text.Document document = new com.itextpdf.text
159             .Document(PageSize.LETTER , 45, 45, 54, 45);

```



```

152 Image imagen = Image.getInstance("./src/es/uco/simas/
153 resources/logo2Antes.png");
154 imagen.setAlignment(com.itextpdf.text.Element.ALIGN_CENTER);
155
156 imagen.scalePercent(40);
157
158 LineSeparator ls = new LineSeparator();
159 BaseFont bf;
160 bf = BaseFont.createFont(font, BaseFont.IDENTITY_H,
161 BaseFont.EMBEDDED);
162
163 Font titulo = new Font(bf, 21, Font.BOLD);
164 Font font2 = new Font(bf, 15, Font.BOLD);
165 Font font3 = new Font(bf, 12);
166 Font font4 = new Font(bf, 12);
167 Font font5 = new Font(bf, 13, Font.BOLD);
168 BaseColor claro = new BaseColor(63,171,160);
169 Font tabla1 = new Font(bf, 12, Font.BOLD);
170 Font azul = new Font(bf, 12);
171 Font rojo = new Font(bf, 12);
172 Font verde = new Font(bf, 12);
173 Font magenta = new Font(bf, 12);
174
175 rojo.setColor(BaseColor.RED);
176 azul.setColor(BaseColor.BLUE);
177 verde.setColor(BaseColor.GREEN);
178 magenta.setColor(BaseColor.MAGENTA);
179 titulo.setColor(33, 77, 72);
180 font2.setColor(BaseColor.BLACK);
181 font3.setColor(43,102,95);
182 ls.setLineWidth(1);
183 ls.setLineColor(claro);
184
185 Paragraph parrafo = new Paragraph();
186 Paragraph met = new Paragraph();
187
188 if(this.metodoSimulacion == 1)
189     parrafo = new Paragraph(" INFORME DE LA SIMULACION
190     DESCENDENTE", titulo);
191
192 if(this.metodoSimulacion == 2){
193     parrafo = new Paragraph(" INFORME DE LA SIMULACION
194     ASCENDENTE", titulo);
195     if(this.metodoAscendente == 0)
196         met = new Paragraph(" METODO SLR", titulo);
197     if(this.metodoAscendente == 1)
198         met = new Paragraph(" METODO LR-CANONICO", titulo);
199     if(this.metodoAscendente == 2)
200         met = new Paragraph(" METODO LALR", titulo);
201 }

```

```

197         int i =0;
198         int j =0;
199         Paragraph parrafo1 = new Paragraph("\n Producciones de la
           gramática: ", font2);
200
201         DefaultListModel produc = this.gramatica.getProducciones();
202         DefaultListModel produc2 = new DefaultListModel();
203         i =0;
204         Object obj;
205         obj = "P {";
206         produc2.addElement(obj);
207         while (i < produc.size()){
208             obj = "      "+(i+1)+"      "+produc.elementAt(i);
209             produc2.addElement(obj);
210             i++;
211         }
212         obj = "}";
213         produc2.addElement(obj);
214
215         i=0;
216         while(i<produc2.getSize()){
217
218             parrafo1.add(new Paragraph("      "+produc2.elementAt(
219                 i).toString(), font4));
220             i++;
221         }
222         Paragraph parrafo2 = new Paragraph("\n Conjunto Primero y
           Siguiente: \n\n", font2);
223         PdfPTable table = new PdfPTable(3);
224
225         PdfPCell celda =new PdfPCell (new Paragraph(" Símbolos",
           tabla1));
226         table.addCell(celda);
227         celda =new PdfPCell (new Paragraph(" Conjunto Primero",
           tabla1));
228         table.addCell(celda);
229         celda =new PdfPCell (new Paragraph(" Conjunto Siguiente",
           tabla1));
230         table.addCell(celda);
231
232         i=0;
233         while(i < this.gramatica.getNoTerm().size()){
234             String primeros = "";
235             String siguientes = "";
236             celda =new PdfPCell (new Paragraph(this.gramatica.
           getNoTerm().get(i).getNombre(), font4));
237             table.addCell(celda);
238             j=0;
239             while(j < this.gramatica.getNoTerm().get(i).getPrimeros
           ().size()){
240                 primeros = primeros + " " + this.gramatica.getNoTerm
           ().get(i).getPrimeros().get(j).getNombre();
           j++;

```

```

241     }
242     celda =new PdfPCell (new Paragraph(primeros , font4));
243     table.addCell(celda);
244     j=0;
245     while(j < this.gramatica.getNoTerm().get(i).
        getSiguientes().size()){
246         siguientes = siguientes + " " + this.gramatica.
            getNoTerm().get(i).getSiguientes().get(j).
            getNombre();
247         j++;
248     }
249     celda =new PdfPCell (new Paragraph(siguientes , font4));
250     table.addCell(celda);
251     i++;
252 }
253
254 Paragraph parrafo3 = new Paragraph("\n Funciones de Error:
    ", font2);
255 if(this.gramatica.getTPredictiva().getFunError().size() ==
    0)
256     parrafo3.add(new Paragraph("No se han declarado
        Funciones de Error.", font4));
257 else{
258     ArrayList<FuncionError> funError = this.gramatica.
        getTPredictiva().getFunError();
259     StringBuilder string = new StringBuilder();
260     j= 0;
261
262     while(j < funError.size()) {
263         string = new StringBuilder();
264         int accion;
265
266         string = string.append(funError.get(j).
            getIdentificador());
267         string = string.append(" - ");
268         accion = funError.get(j).getAccion();
269         if(accion == 1)
270             string.append("Insertar un SÃmbolo en la
                Entrada: ");
271         if(accion == 2)
272             string.append("Borrar un SÃmbolo de la Entrada
                ");
273         if(accion == 3)
274             string.append("Modificar un SÃmbolo de la
                Entrada: ");
275         if(accion == 4)
276             string.append("Insertar un SÃmbolo de la Pila:
                ");
277         if(accion == 5)
278             string.append("Borrar un SÃmbolo de la Pila");
279         if(accion == 6)
280             string.append("Modificar un SÃmbolo de la Pila
                : ");

```

```

281         if(accion == 7)
282             string.append("Terminar el an lisis");
283         if(accion == 1 || accion ==3 || accion ==4 ||
           accion ==6)
284             string.append(funError.get(j).getSimbolo().
                           getNombre());
285
286         parrafo3.add(new Paragraph("\n      "+string, font4))
           ;
287         j++;
288     }
289 }
290 Paragraph parrafo4 = new Paragraph();
291 if(this.cadenaEntrada != null){
292     parrafo4 = new Paragraph("\n Cadena de Entrada: ",
           font2);
293     i = 0;
294     String str = " ";
295     while(i < this.cadenaEntrada.size()){
296         str = str + " "+this.cadenaEntrada.get(i).getNombre
           ();
297         i++;
298     }
299     parrafo4.add(new Paragraph(" "+str, font4));
300 }
301 Paragraph parrafo5 = new Paragraph();
302 PdfPTable table2 = new PdfPTable(1);
303 Paragraph parrafo6 = new Paragraph();
304 PdfPTable table3 = new PdfPTable(1);
305 PdfPTable table4 = new PdfPTable(1);
306 Paragraph col1 = new Paragraph("\n Coleccion Canonica
           Elementos LR(0) \n\n", font2);
307 PdfPTable col2 = new PdfPTable(1);
308 Paragraph accion = new Paragraph("      PARTE ACCION \n\n",
           font5);
309 Paragraph ira = new Paragraph("      PARTE IR_A \n\n", font5
           );
310
311 if(this.metodoSimulacion == 1){ // Metodo Descendente
312     parrafo5 = new Paragraph("\n Tabla Predictiva: \n\n",
           font2);
313     DefaultTableModel tpredictiva = this.gramatica.
           getTPredictiva().getTabla();
314     table2 = new PdfPTable(tpredictiva.getColumnCount());
315     i=0;
316     while(i < tpredictiva.getColumnCount()){
317         celda =new PdfPCell (new Paragraph(tpredictiva.
           getColumnName(i), tabla1));
318         table2.addCell(celda);
319         i++;
320     }
321     i =0;
322     j =0;

```

```

323         while(i < tpredictiva.getRowCount()){
324             j=0;
325             while(j < tpredictiva.getColumnCount()){
326                 if(tpredictiva.getValueAt(i, j) == null){
327                     table2.addCell("");
328                 }else{
329                     if(tpredictiva.getValueAt(i, j).toString().
                        startsWith("<")){
330                         celda =new PdfPCell (new Paragraph("
Emparejar", azul));
331                         table2.addCell(celda);
332                     }else{
333                         if(j==0)
334                             celda =new PdfPCell (new Paragraph(
                                tpredictiva.getValueAt(i, j).
                                toString(), tabla1));
335                     else{
336                         if(tpredictiva.getValueAt(i, j).
                                toString().startsWith("E"))
337                             celda =new PdfPCell (new
                                Paragraph(tpredictiva.
                                getValueAt(i, j).toString()
                                , rojo));
338                         else
339                             celda =new PdfPCell (new
                                Paragraph(tpredictiva.
                                getValueAt(i, j).toString()
                                , font4));
340                     }
341                     table2.addCell(celda);
342                 }
343             }
344             j++;
345         }
346         i++;
347     }
348     if(this.cadenaEntrada != null){
349         parrafo6 = new Paragraph("\n Simulacion Descendente
: \n\n", font2);
350         DefaultTableModel sim = this.gramatica.
            getTPredictiva().getTabla();
351         NuevaSimulacionDesc nuevo = new NuevaSimulacionDesc
            (this.gramatica, this);
352         sim = nuevo.getTabla();
353         table3 = new PdfPTable(sim.getColumnCount());
354         i=0;
355         while(i < sim.getColumnCount()){
356             celda =new PdfPCell (new Paragraph(sim.
                getColumnName(i), tabla1));
357             table3.addCell(celda);
358             i++;
359         }
360     }

```

```

361         i =0;
362         j =0;
363         while(i < sim.getRowCount()){
364             j=0;
365             while(j < sim.getColumnCount()){
366                 celda =new PdfPCell (new Paragraph(sim.
                    getValueAt(i, j).toString(), font4));
367                 table3.addCell(celda);
368                 j++;
369             }
370             i++;
371         }
372     }
373 }
374 if (this.metodoSimulacion == 2){ //Metodo ascendente
375     parrafo5 = new Paragraph("\n Tabla LR: \n\n", font2);
376     DefaultTableModel taccion = this.gramatica.getTlr().
        getTaccion().getMatrizAccion();
377     DefaultTableModel tira = this.gramatica.getTlr().
        getTlrA().getMatrizIrA();
378     table2 = new PdfPTable(taccion.getColumnCount());
379     table4 = new PdfPTable(tira.getColumnCount()+1);
380     if (this.metodoAscendente ==0)
381         celda =new PdfPCell (new Paragraph(this.gramatica.
            getColeccionLR0().getColeccion(), font4));
382     if (this.metodoAscendente ==1)
383         celda =new PdfPCell (new Paragraph(this.gramatica.
            getColeccionLR1().getColeccion(), font4));
384     col2.addCell(celda);
385     i=0;
386     while(i < taccion.getColumnCount()){ //parte accion
387         celda = new PdfPCell (new Paragraph(taccion.
            getColumnName(i), tabla1));
388         table2.addCell(celda);
389         i++;
390     }
391     i=0;
392     celda = new PdfPCell (new Paragraph(taccion.
        getColumnName(i), tabla1));
393     table4.addCell(celda);
394     i=0;
395     while(i < tira.getColumnCount()){ // parte ir_a
396         celda =new PdfPCell (new Paragraph(tira.
            getColumnName(i), tabla1));
397         table4.addCell(celda);
398         i++;
399     }
400     i =0;
401     j =0;
402     while(i < taccion.getRowCount()){ //parte accion
403         j=0;
404         while(j < taccion.getColumnCount()){
405             if (taccion.getValueAt(i, j) == null){

```

```

406         table2.addCell("");
407     }else{
408         if(taccion.getValueAt(i, j).toString().
409             equals("Aceptar")){
410             celda =new PdfPCell (new Paragraph("
411                 Aceptar", verde));
412             table2.addCell(celda);
413         }else{
414             if(taccion.getValueAt(i, j).toString().
415                 startsWith("<")){
416                 celda =new PdfPCell (new Paragraph(
417                     "Emparejar", azul));
418                 table2.addCell(celda);
419             }else{
420                 if(j==0)
421                     celda =new PdfPCell (new
422                         Paragraph(taccion.
423                             getValueAt(i, j).toString()
424                                 , tabla1));
425                 else{
426                     if(taccion.getValueAt(i, j).
427                         toString().startsWith("E"))
428                     {
429                         celda =new PdfPCell (new
430                             Paragraph(taccion.
431                                 getValueAt(i, j).
432                                     toString(), rojo));
433                     }else{
434                         if(taccion.getValueAt(i, j)
435                             .toString().startsWith(
436                                 "d")){
437                             celda =new PdfPCell (
438                                 new Paragraph(
439                                     taccion.getValueAt(
440                                         i, j).toString(),
441                                         azul));
442                         }else{
443                             if(taccion.getValueAt(i
444                                 , j).toString().
445                                 startsWith("r")){
446                                 celda =new PdfPCell
447                                     (new Paragraph
448                                         (taccion.
449                                             getValueAt(i, j
450                                                 ).toString(),
451                                                 magenta));
452                             }else{
453                                 if(taccion.
454                                     getValueAt(i, j
455                                         ).toString().
456                                         startsWith("
457                                             conf")){

```





```

462         celda =new PdfPCell (new Paragraph(tira
                                .getValueAt(i, j).toString(), font4
                                ));
463
464         table4.addCell(celda);
465
466     }
467     j++;
468 }
469 i++;
470 }
471
472 if(this.cadenaEntrada != null){
473     parrafo6 = new Paragraph("\n Simulacion Ascendente:
                                \n\n", font2);
474     DefaultTableModel sim = this.gramatica.
                                getTPredictiva().getTabla();
475     NuevaSimulacionAsc nuevo = new NuevaSimulacionAsc(
                                this.gramatica, this);
476     sim = nuevo.getTabla();
477     table3 = new PdfPTable(sim.getColumnCount());
478     i=0;
479     while(i < sim.getColumnCount()){
480         celda =new PdfPCell (new Paragraph(sim.
                                getColumnName(i), tabla1));
481         table3.addCell(celda);
482         i++;
483     }
484
485     i =0;
486     j =0;
487     while(i < sim.getRowCount()){
488         j=0;
489         while(j < sim.getColumnCount()){
490             celda =new PdfPCell (new Paragraph(sim.
                                getValueAt(i, j).toString(), font4));
491             table3.addCell(celda);
492             j++;
493         }
494         i++;
495     }
496 }
497 }
498 try {
499
500     PdfWriter.getInstance(document, new FileOutputStream(
                                fichero));
501 } catch (DocumentException | FileNotFoundException ex) {
502     Logger.getLogger(Editor.class.getName()).log(Level.
                                SEVERE, null, ex);
503 }
504
505 document.open();

```

```

506         try {
507
508             document.add(imagen);
509             document.add(parrafo);
510             if (this.metodoSimulacion == 2)
511                 document.add(met);
512
513             document.add(new Chunk(ls));
514             document.add(parrafo1);
515             document.add(parrafo2);
516             document.add(table);
517             document.add(parrafo3);
518             document.add(parrafo4);
519             if (this.metodoSimulacion == 2){
520                 document.add(col1);
521                 document.add(col2);
522             }
523             document.add(parrafo5);
524             if (this.metodoSimulacion == 2)
525                 document.add(accion);
526             document.add(table2);
527             if (this.metodoSimulacion == 2){
528                 document.add(ira);
529                 document.add(table4);
530             }
531             document.add(parrafo6);
532             document.add(table3);
533
534         } catch (DocumentException ex) {
535             Logger.getLogger(Editor.class.getName()).log(Level.
                    SEVERE, null, ex);
536         }
537
538         document.close();
539     } catch (BadElementException ex) {
540         Logger.getLogger(Editor.class.getName()).log(Level.SEVERE,
                    null, ex);
541     } catch (MalformedURLException ex) {
542         Logger.getLogger(Editor.class.getName()).log(Level.SEVERE,
                    null, ex);
543     } catch (IOException ex) {
544         Logger.getLogger(Editor.class.getName()).log(Level.SEVERE,
                    null, ex);
545     }
546
547     return true;
548 }
549
550 /**
551  * This method is called from within the constructor to initialize
552  * the form.
553  * WARNING: Do NOT modify this code. The content of this method is
554  * always

```

```

553      * regenerated by the Form Editor.
554      */
555      @SuppressWarnings("unchecked")
556      // <editor-fold defaultstate="collapsed" desc="Generated Code">//
          GEN-BEGIN: initComponents
557      private void initComponents() {
558
559          jScrollPane3 = new javax.swing.JScrollPane();
560          jTable1 = new javax.swing.JTable();
561          jPanel1 = new javax.swing.JPanel();
562          jLabelSimulador = new javax.swing.JLabel();
563          jScrollPane1 = new javax.swing.JScrollPane();
564          jList1 = new javax.swing.JList();
565          jLabel2 = new javax.swing.JLabel();
566          jScrollPane2 = new javax.swing.JScrollPane();
567          jList2 = new javax.swing.JList();
568          jLabel3 = new javax.swing.JLabel();
569          jScrollPane4 = new javax.swing.JScrollPane();
570          jTable2 = new javax.swing.JTable();
571          jButton1 = new javax.swing.JButton();
572          jScrollPane5 = new javax.swing.JScrollPane();
573          jTable3 = new javax.swing.JTable();
574          jButton4 = new javax.swing.JButton();
575          jToolBar1 = new javax.swing.JToolBar();
576          jButton3 = new javax.swing.JButton();
577          jSeparator1 = new javax.swing.JToolBar.Separator();
578          jButton2 = new javax.swing.JButton();
579          jLabelTabla = new javax.swing.JLabel();
580          jLabelAccion = new javax.swing.JLabel();
581          jLabelIra = new javax.swing.JLabel();
582          jMenuBar1 = new javax.swing.JMenuBar();
583          jMenu1 = new javax.swing.JMenu();
584          jMenuItem1 = new javax.swing.JMenuItem();
585          jMenuItem2 = new javax.swing.JMenuItem();
586          jMenu2 = new javax.swing.JMenu();
587          jMenuItem3 = new javax.swing.JMenuItem();
588          jMenuItem4 = new javax.swing.JMenuItem();
589
590          jTable1.setModel(new javax.swing.table.DefaultTableModel(
591              new Object [][] {
592                  {null, null, null, null},
593                  {null, null, null, null},
594                  {null, null, null, null},
595                  {null, null, null, null}
596              },
597              new String [] {
598                  "Title 1", "Title 2", "Title 3", "Title 4"
599              }
600          ));
601          jScrollPane3.setViewportView(jTable1);
602
603          setDefaultCloseOperation(javax.swing.WindowConstants.
              DISPOSE_ON_CLOSE);

```

```

604 setTitle("Simulador");
605
606 jPanel1.setBackground(new java.awt.Color(233, 244, 244));
607 jPanel1.setToolTipText("");
608 jPanel1.setCursor(new java.awt.Cursor(java.awt.Cursor.
        DEFAULT_CURSOR));
609 jPanel1.setFocusable(false);
610
611 jLabelSimulador.setFont(new java.awt.Font("Tahoma", 1, 18)); //
        NOI18N
612 jLabelSimulador.setText("jLabel1");
613
614 jList1.setPreferredSize(new java.awt.Dimension(706, 770));
615 jScrollPane1.setViewportViewView(jList1);
616
617 jLabel2.setFont(new java.awt.Font("Ubuntu", 1, 15)); // NOI18N
618 jLabel2.setText("Gramática");
619
620 jScrollPane2.setViewportViewView(jList2);
621
622 jLabel3.setFont(new java.awt.Font("Ubuntu", 1, 15)); // NOI18N
623 jLabel3.setText("Funciones de Error");
624
625 jTable2.setModel(new javax.swing.table.DefaultTableModel(
626     new Object [][] {
627         {null, null, null, null},
628         {null, null, null, null},
629         {null, null, null, null},
630         {null, null, null, null}
631     },
632     new String [] {
633         "", "", "", ""
634     }
635 ) {
636     boolean[] canEdit = new boolean [] {
637         false, false, false, false
638     };
639
640     public boolean isCellEditable(int rowIndex, int columnIndex
641         ) {
642         return canEdit [columnIndex];
643     }
644 });
645 jScrollPane4.setViewportViewView(jTable2);
646
647 jButton1.setText("Modificar Funciones de Error");
648 jButton1.addActionListener(new java.awt.event.ActionListener()
649 {
650     public void actionPerformed(java.awt.event.ActionEvent evt)
651     {
652         jButton1ActionPerformed(evt);
653     }
654 });

```

```

652
653     jTable3.setModel(new javax.swing.table.DefaultTableModel(
654         new Object [][] {
655             {},
656             {},
657             {},
658             {}
659         },
660         new String [] {
661
662         }
663     ));
664     jScrollPane5.setViewportViewView(jTable3);
665
666     jButton4.setText("Generar Informe");
667     jButton4.addActionListener(new java.awt.event.ActionListener()
668     {
669         public void actionPerformed(java.awt.event.ActionEvent evt)
670         {
671             jButton4ActionPerformed(evt);
672         }
673     });
674
675     jToolBar1.setRollover(true);
676
677     jButton3.setFont(new java.awt.Font("Ubuntu", 1, 15)); // NOI18N
678     jButton3.setText("Similar");
679     jButton3.setFocusable(false);
680     jButton3.setHorizontalTextPosition(javax.swing.SwingConstants.
        CENTER);
681     jButton3.setVerticalTextPosition(javax.swing.SwingConstants.
        BOTTOM);
682     jButton3.addActionListener(new java.awt.event.ActionListener()
683     {
684         public void actionPerformed(java.awt.event.ActionEvent evt)
685         {
686             jButton3ActionPerformed(evt);
687         }
688     });
689     jToolBar1.add(jButton3);
690     jToolBar1.add(jSeparator1);
691
692     jButton2.setIcon(new javax.swing.ImageIcon(getClass().
        getResource("/es/uco/simas/resources/salir.png"))); //
        NOI18N
693     jButton2.setToolTipText("Salir");
694     jButton2.setFocusable(false);
695     jButton2.setHorizontalTextPosition(javax.swing.SwingConstants.
        CENTER);
696     jButton2.setVerticalTextPosition(javax.swing.SwingConstants.
        BOTTOM);
697     jButton2.addActionListener(new java.awt.event.ActionListener()
698     {

```

```

694         public void actionPerformed(java.awt.event.ActionEvent evt)
695         {
696             jButton2ActionPerformed(evt);
697         }
698     });
699     jToolBar1.add(jButton2);
700     jLabelTabla.setFont(new java.awt.Font("Ubuntu", 1, 15)); //
701     //NOI18N
702     jLabelTabla.setText("jLabel1");
703     jLabelAccion.setFont(new java.awt.Font("Ubuntu", 1, 15)); //
704     //NOI18N
705     jLabelAccion.setText("Parte Accion");
706     jLabelIra.setFont(new java.awt.Font("Ubuntu", 1, 15)); //
707     //NOI18N
708     jLabelIra.setText("Parte Ir_a");
709     javax.swing.GroupLayout jPanel1Layout = new javax.swing.
710     GroupLayout(jPanel1);
711     jPanel1.setLayout(jPanel1Layout);
712     jPanel1Layout.setHorizontalGroup(
713     jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
714     Alignment.LEADING)
715     .addGroup(jPanel1Layout.createSequentialGroup()
716     .addGroup(jPanel1Layout.createParallelGroup(javax.swing.
717     GroupLayout.Alignment.LEADING)
718     .addGroup(jPanel1Layout.createSequentialGroup()
719     .addComponent(jScrollPane4, javax.swing.
720     GroupLayout.PREFERRED_SIZE, 465,
721     javax.swing.GroupLayout.
722     PREFERRED_SIZE)
723     .addPreferredGap(javax.swing.
724     GroupLayoutStyle.ComponentPlacement.
725     RELATED)
726     .addComponent(jScrollPane5, javax.swing.
727     GroupLayout.PREFERRED_SIZE, 325,
728     javax.swing.GroupLayout.
729     PREFERRED_SIZE))
730     .addGroup(jPanel1Layout.createSequentialGroup()
731     .addComponent(jScrollPane1, javax.
732     swing.GroupLayout.
733     PREFERRED_SIZE, 346, javax.

```

```

725         swing.GroupLayout.
            PREFERRED_SIZE)
726     .addGroup(jPanel1Layout.
727         createSequentialGroup())
728         .addGap(48, 48, 48)
729         .addComponent(jLabelTabla)
730         .addGap(109, 109, 109)
731         .addComponent(jLabelAccion)))
732     .addGap(18, 18, 18)
733     .addGroup(jPanel1Layout.
734         createParallelGroup(javax.swing.
735             GroupLayout.Alignment.LEADING)
736         .addComponent(jScrollPane2, javax.
737             swing.GroupLayout.
738                 PREFERRED_SIZE, 282, javax.
739                 swing.GroupLayout.
740                 PREFERRED_SIZE)
741         .addGroup(javax.swing.GroupLayout.
742             Alignment.TRAILING,
743             jPanel1Layout.
744                 createSequentialGroup()
745                 .addComponent(jLabelIra)
746                 .addGap(38, 38, 38))))
747     .addGroup(jPanel1Layout.
748         createSequentialGroup()
749         .addGap(3, 3, 3)
750         .addComponent(jLabel2)
751         .addGap(301, 301, 301)
752         .addComponent(jLabel3))))
753     .addGroup(jPanel1Layout.createSequentialGroup())
754     .addGap(174, 174, 174)
755     .addComponent(jButton1))
756     .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
757         Short.MAX_VALUE))
758     .addGroup(jPanel1Layout.createSequentialGroup())
759     .addGap(283, 283, 283)
760     .addComponent(jLabelSimulador)
761     .addPreferredGap(javax.swing.LayoutStyle.
762         ComponentPlacement.RELATED, javax.swing.GroupLayout.
763             DEFAULT_SIZE, Short.MAX_VALUE)
764     .addComponent(jButton4)
765     .addGap(113, 113, 113))
766     .addGroup(jPanel1Layout.createSequentialGroup())
767     .addComponent(jToolBar1, javax.swing.GroupLayout.
768         DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
769         Short.MAX_VALUE)
770     .addContainerGap())
771 );
772 jPanel1Layout.setVerticalGroup(
773     jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
774         Alignment.LEADING)
775     .addGroup(jPanel1Layout.createSequentialGroup())

```

```

758 .addComponent(jToolBar1 , javax.swing.GroupLayout .
      PREFERRED_SIZE, 34, javax.swing.GroupLayout .
      PREFERRED_SIZE)
759 .addGap(18, 18, 18)
760 .addGroup(jPanel1Layout.createParallelGroup(javax.swing
      .GroupLayout.Alignment.BASELINE)
761 .addComponent(jLabelSimulador)
762 .addComponent(jButton4))
763 .addGap(26, 26, 26)
764 .addGroup(jPanel1Layout.createParallelGroup(javax.swing
      .GroupLayout.Alignment.BASELINE)
765 .addComponent(jLabel2)
766 .addComponent(jLabel3))
767 .addGap(18, 18, 18)
768 .addGroup(jPanel1Layout.createParallelGroup(javax.swing
      .GroupLayout.Alignment.LEADING)
769 .addComponent(jScrollPane2 , javax.swing.GroupLayout
      .PREFERRED_SIZE, 219, javax.swing.GroupLayout .
      PREFERRED_SIZE)
770 .addGroup(jPanel1Layout.createSequentialGroup())
771 .addComponent(jScrollPane1 , javax.swing .
      GroupLayout.PREFERRED_SIZE, 209, javax .
      swing.GroupLayout.PREFERRED_SIZE)
772 .addGroup(jPanel1Layout.createParallelGroup(
      javax.swing.GroupLayout.Alignment.LEADING)
773 .addGroup(jPanel1Layout .
      createSequentialGroup())
774 .addPreferredGap(javax.swing .
      LayoutStyle.ComponentPlacement .
      RELATED)
775 .addComponent(jLabelTabla))
776 .addGroup(jPanel1Layout .
      createSequentialGroup())
777 .addGap(20, 20, 20)
778 .addGroup(jPanel1Layout .
      createParallelGroup(javax.swing .
      GroupLayout.Alignment.BASELINE)
779 .addComponent(jLabelAccion)
780 .addComponent(jLabelIra))))))
781 .addPreferredGap(javax.swing.LayoutStyle .
      ComponentPlacement.RELATED)
782 .addGroup(jPanel1Layout.createParallelGroup(javax.swing
      .GroupLayout.Alignment.LEADING, false)
783 .addComponent(jScrollPane5 , javax.swing.GroupLayout
      .DEFAULT_SIZE, 254, Short.MAX_VALUE)
784 .addComponent(jScrollPane4 , javax.swing.GroupLayout
      .PREFERRED_SIZE, 0, Short.MAX_VALUE))
785 .addPreferredGap(javax.swing.LayoutStyle .
      ComponentPlacement.RELATED)
786 .addComponent(jButton1)
787 .addContainerGap(136, Short.MAX_VALUE))
788 );
789

```



```
790     jMenuItem1.setText("Simulador");
791
792     jMenuItem1.setText("Nueva Simulacion");
793     jMenuItem1.add(jMenuItem1);
794
795     jMenuItem2.setText("Salir");
796     jMenuItem2.addActionListener(new java.awt.event.ActionListener
797         () {
798         public void actionPerformed(java.awt.event.ActionEvent evt)
799             {
800             jButton2ActionPerformed(evt);
801         }
802     });
803     jMenuItem1.add(jMenuItem2);
804
805     jMenuItemBar1.add(jMenuItem1);
806
807     jMenuItem2.setText("Ayuda");
808
809     jMenuItem3.setText("Centro de Ayuda");
810     jMenuItem3.addActionListener(new java.awt.event.ActionListener
811         () {
812         public void actionPerformed(java.awt.event.ActionEvent evt)
813             {
814             jMenuItem3ActionPerformed(evt);
815         }
816     });
817     jMenuItem2.add(jMenuItem3);
818
819     jMenuItem4.setText("Acerca de ...");
820     jMenuItem4.addActionListener(new java.awt.event.ActionListener
821         () {
822         public void actionPerformed(java.awt.event.ActionEvent evt)
823             {
824             jMenuItem4ActionPerformed(evt);
825         }
826     });
827     jMenuItem2.add(jMenuItem4);
828
829     jMenuItemBar1.add(jMenuItem2);
830
831     setJMenuBar(jMenuBar1);
832
833     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
834         getContentPane());
835     getContentPane().setLayout(layout);
836     layout.setHorizontalGroup(
837         layout.createParallelGroup(javax.swing.GroupLayout.
838             Alignment.LEADING)
839         .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
840             layout.createSequentialGroup()
841                 .addComponent(jPanel1, javax.swing.GroupLayout.
842                     PREFERRED_SIZE, javax.swing.GroupLayout.
```

```

            DEFAULT.SIZE, javax.swing.GroupLayout.
            PREFERRED.SIZE)
833         .addGap(0, 0, Short.MAX.VALUE))
834     );
835     layout.setVerticalGroup(
836         layout.createParallelGroup(javax.swing.GroupLayout.
            Alignment.LEADING)
837         .addGroup(layout.createSequentialGroup())
838         .addContainerGap()
839         .addComponent(jPanel1, javax.swing.GroupLayout.
            DEFAULT.SIZE, javax.swing.GroupLayout.DEFAULT.SIZE,
            Short.MAX.VALUE)
            .addContainerGap());
840
841
842
843     pack();
844 }// </editor-fold>//GEN-END: initComponents
845
846 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt
847 ) { //GEN-FIRST: event_jButton2ActionPerformed
848     int conf = JOptionPane.showConfirmDialog(null, "¿Desea salir
849     del simulador?", "Salir", JOptionPane.YES_NO_OPTION);
850
851     if (conf == 0) {
852         this.dispose();
853     }
854 } //GEN-LAST: event_jButton2ActionPerformed
855
856 private void jButton3ActionPerformed(java.awt.event.ActionEvent evt
857 ) { //GEN-FIRST: event_jButton3ActionPerformed
858     if (this.metodoSimulacion == 1) {
859         NuevaSimulacionDesc simDesc = new NuevaSimulacionDesc(this.
860             gramatica, this);
861         simDesc.setVisible(true);
862         simDesc.setLocationRelativeTo(null);
863     }
864     if (this.metodoSimulacion == 2) {
865         NuevaSimulacionAsc simAsc = new NuevaSimulacionAsc(this.
866             gramatica, this);
867         simAsc.setVisible(true);
868         simAsc.setLocationRelativeTo(null);
869     }
870 } //GEN-LAST: event_jButton3ActionPerformed
871
872 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt
873 ) { //GEN-FIRST: event_jButton1ActionPerformed
874     if (this.metodoSimulacion == 1) {
875         new VentanaSimuladorDesc(this.gramatica).cambiarPaso(4);
876         this.dispose();
877     }
878     if (this.metodoSimulacion == 2) {

```

```

875         new VentanaSimuladorAsc(this.gramatica).cambiarPaso(5);
876         this.dispose();
877     }
878 }//GEN-LAST:event_jButton1ActionPerformed
879
880 private void jButton4ActionPerformed(java.awt.event.ActionEvent evt
881 ) { //GEN-FIRST:event_jButton4ActionPerformed
882
883     Boolean resultado= null;
884     FileNameExtensionFilter filtro= null;
885     JFileChooser selector= null;
886
887     JFileChooser chooser = new JFileChooser();
888     selector=chooser;
889     FileNameExtensionFilter extension = new FileNameExtensionFilter
890         ("Informes de simulacion Ascendente (.pdf)",new String[]
891         {"pdf"});
892     filtro=extension;
893     selector.setFileFilter(filtro);
894     File fichero= new File("");
895     if(this.metodoSimulacion==1)
896         fichero = new File("informeSimulacionDesc.pdf");
897     if(this.metodoSimulacion==2)
898         fichero = new File("informeSimulacionAsc.pdf");
899     selector.setSelectedFile(fichero);
900     if(selector.showSaveDialog(null)==0) {
901         try {
902             resultado = this.generarInforme(selector.
903                 getSelectedFile().toString());
904         } catch (DocumentException ex) {
905             Logger.getLogger(Editor.class.getName()).log(Level.
906                 SEVERE, null, ex);
907         }
908     }
909
910     if(resultado.booleanValue()) {
911         StringBuilder JdecGenerated80 = new StringBuilder();
912     } else {
913         JOptionPane.showConfirmDialog(null,"El informe de la
914             gramática no se puede generar hasta que la
915             gramática esté validada.", "Informe de la
916             gramática", JOptionPane.DEFAULT_OPTION);
917     }
918 }
919 }//GEN-LAST:event_jButton4ActionPerformed
920
921 private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent
922     evt) { //GEN-FIRST:event_jMenuItem3ActionPerformed
923     // TODO add your handling code here:
924 }//GEN-LAST:event_jMenuItem3ActionPerformed
925

```

```

919     private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent
920         evt) {//GEN-FIRST:event_jMenuItem4ActionPerformed
921         AcercaDe acerca = new AcercaDe();
922         acerca.setVisible(true);
923         acerca.setLocationRelativeTo(null);
924     }//GEN-LAST:event_jMenuItem4ActionPerformed
925
926     public Gramatica getGramatica() {
927         return this.gramatica;
928     }
929
930     public ArrayList<Terminal> getCadenaEntrada() {
931         return cadenaEntrada;
932     }
933
934     public void setCadenaEntrada(ArrayList<Terminal> cadenaEntrada) {
935         this.cadenaEntrada = cadenaEntrada;
936     }
937
938     public int getMetodoSimulacion() {
939         return this.metodoSimulacion;
940     }
941
942     public void setMetodoSimulacion(int metodoSimulacion) {
943         this.metodoSimulacion = metodoSimulacion;
944     }
945
946     public int getModoFuncionamiento() {
947         return modoFuncionamiento;
948     }
949
950     public void setModoFuncionamiento(int modoFuncionamiento) {
951         this.modoFuncionamiento = modoFuncionamiento;
952     }
953
954     public int getMetodoAscendente() {
955         return this.metodoAscendente;
956     }
957
958     public void setMetodoAscendente(int metodoAscendente) {
959         this.metodoAscendente = metodoAscendente;
960     }
961
962     public String actualizarVisualizacion() {
963         String cadena = "";
964         ArrayList<Terminal> cadenaEntrada = this.getCadenaEntrada();
965         int i = 0;
966         while (i < cadenaEntrada.size()) {
967             cadena = cadena+cadenaEntrada.get(i).getNombre()+" ";
968             i++;
969         }
970         return cadena;

```

```

971
972 // Variables declaration – do not modify//GEN-BEGIN:variables
973 private javax.swing.JButton jButton1;
974 private javax.swing.JButton jButton2;
975 private javax.swing.JButton jButton3;
976 private javax.swing.JButton jButton4;
977 private javax.swing.JLabel jLabel2;
978 private javax.swing.JLabel jLabel3;
979 private javax.swing.JLabel jLabelAccion;
980 private javax.swing.JLabel jLabelIra;
981 private javax.swing.JLabel jLabelSimulador;
982 private javax.swing.JLabel jLabelTabla;
983 private javax.swing.JList jList1;
984 private javax.swing.JList jList2;
985 private javax.swing.JMenu jMenuItem1;
986 private javax.swing.JMenu jMenuItem2;
987 private javax.swing.JMenuBar jMenuItemBar1;
988 private javax.swing.JMenuItem jMenuItem1;
989 private javax.swing.JMenuItem jMenuItem2;
990 private javax.swing.JMenuItem jMenuItem3;
991 private javax.swing.JMenuItem jMenuItem4;
992 private javax.swing.JPanel jPanel1;
993 private javax.swing.JScrollPane jScrollPane1;
994 private javax.swing.JScrollPane jScrollPane2;
995 private javax.swing.JScrollPane jScrollPane3;
996 private javax.swing.JScrollPane jScrollPane4;
997 private javax.swing.JScrollPane jScrollPane5;
998 private javax.swing.JToolBar.Separator jSeparator1;
999 private javax.swing.JTable jTable1;
1000 private javax.swing.JTable jTable2;
1001 private javax.swing.JTable jTable3;
1002 private javax.swing.JToolBar jToolBar1;
1003 // End of variables declaration//GEN-END:variables
1004 }

```

### 2.5.18. VentanaSimuladorAsc.java

```

1 //SimAS / Simulador
2 //Ventana simulador Ascendente
3
4 package es.uco.simas.simulador;
5
6 import es.uco.simas.util.gramatica.Gramatica;
7 import es.uco.simas.SimAS;
8 import es.uco.simas.editor.ColCanLR0;
9 import es.uco.simas.editor.ColCanLR1;
10 import es.uco.simas.editor.ColCanLALR;
11
12 /**
13  * @author vanesa

```

```
14  */
15  public class VentanaSimuladorAsc extends javax.swing.JFrame {
16
17      private PanelNuevaSimAscPaso1 paso1;
18      PanelNuevaSimAscPaso2 paso2;
19      private PanelNuevaSimAscPaso3 paso3;
20      private PanelNuevaSimAscPaso4 paso4;
21      private PanelNuevaSimAscPaso5 paso5;
22      private PanelNuevaSimAscPaso6 paso6;
23
24      public Simulador simulacion ;
25      public Gramatica gramatica;
26      public int metodo = -1;
27
28      public VentanaSimuladorAsc(Gramatica gramatica) {
29          initComponents();
30          this.gramatica = gramatica;
31          this.setResizable(false);
32          PanelNuevaSimAscPaso1 paso1 = new PanelNuevaSimAscPaso1(this);
33          this.paso1 = paso1;
34
35          PanelNuevaSimAscPaso2 paso2 = new PanelNuevaSimAscPaso2(this);
36          this.paso2 = paso2;
37
38          this.getContentPane().removeAll();
39          this.setContentPane(this.paso1);
40          this.pack();
41          this.validate();
42          this.setTitle("Simulador Ascendente. Paso 1 de 5");
43          this.setVisible(true);
44          this.setLocationRelativeTo(null);
45
46      }
47
48      public Simulador getSimulador(){
49          return this.simulacion;
50      }
51      public void setSimulador(Simulador sim){
52          this.simulacion = sim;
53      }
54      public void setMetodo(int m){
55          this.metodo = m;
56      }
57      public int getMetodo(){
58          return this.metodo;
59      }
60
61      public void cambiarPaso( int paso) {
62          switch(paso){
63              case 1:{
64                  this.setContentPane(this.paso1);
65                  this.pack();
66                  this.setVisible(true);
```

```
67         this.validate();
68         this.setTitle("Simulador Ascendente. Paso 1 de 5");
69         break;
70     }
71     case 2: {
72         this.setContentPane(this.paso2);
73         this.pack();
74         this.setVisible(true);
75         this.validate();
76         this.gramatica.generarConjPrim();
77         this.gramatica.generarConjSig();
78         this.paso2.construirConjuntos(this.gramatica);
79         this.setTitle("Simulador Ascendente. Paso 2 de 5");
80         this.setLocationRelativeTo(null);
81         break;
82     }
83     case 3:
84     {
85         PanelNuevaSimAscPaso3 paso3 = new PanelNuevaSimAscPaso3(this);
86         this.paso3 = paso3;
87         this.setContentPane(this.paso3);
88         this.pack();
89         this.setVisible(true);
90         this.validate();
91         this.setTitle("Simulador Ascendente. Paso 3 de 5");
92         break;
93     }
94     case 4:
95     {
96         PanelNuevaSimAscPaso4 paso4 = new PanelNuevaSimAscPaso4(this);
97         this.paso4 = paso4;
98         this.setContentPane(this.paso4);
99         this.pack();
100        this.setVisible(true);
101        this.validate();
102        this.setTitle("Simulador Ascendente. Paso 4 de 5");
103        break;
104    }
105    case 5:
106    {
107        PanelNuevaSimAscPaso5 paso5 = new PanelNuevaSimAscPaso5(this);
108        this.paso5 = paso5;
109        this.setContentPane(this.paso5);
110        this.pack();
111        this.setVisible(true);
112        this.validate();
113        this.setTitle("Simulador Ascendente. Paso 5 de 5");
114        this.setLocationRelativeTo(null);
115        break;
116    }
117    case 6:
118    {
119        PanelNuevaSimAscPaso6 paso6 = new PanelNuevaSimAscPaso6(this);
```

```

120         this.paso6 = paso6;
121         this.setContentPane(this.paso6);
122         this.pack();
123         this.setVisible(true);
124         this.validate();
125         this.setTitle("Simulador Ascendente. Paso 6 de 6");
126         this.setLocationRelativeTo(null);
127         break;
128     }
129 }
130 }
131 public Gramatica getGramatica(){
132     return this.gramatica;
133 }
134
135 void setGramatica(Gramatica gr){
136     this.gramatica = gr;
137 }
138
139 public void finalizarAsistente(){
140     this.dispose();
141     SimAS simas = new SimAS();
142     simas.lanzarSimulador(2, this.gramatica, this.getMetodo());
143 }
144
145 public void setColeccion(ColCanLR0 col){
146     this.gramatica.setColeccionLR0(col);
147 }
148
149 public void setColeccion(ColCanLR1 col){
150     this.gramatica.setColeccionLR1(col);
151 }
152
153 public void setColeccion(ColCanLALR col){
154     this.gramatica.setColeccionLALR(col);
155 }
156
157 /**
158  * This method is called from within the constructor to initialize
159  * the form.
160  * WARNING: Do NOT modify this code. The content of this method is
161  * always
162  * regenerated by the Form Editor.
163  */
164 @SuppressWarnings("unchecked")
165 // <editor-fold defaultstate="collapsed" desc="Generated Code">
166 // GEN-BEGIN: initComponents
167 private void initComponents() {
168     setDefaultCloseOperation(javax.swing.WindowConstants.
        DISPOSE_ON_CLOSE);
    setBounds(new java.awt.Rectangle(0, 0, 0, 0));
    setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));

```



```

169
170     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
171         getContentPane());
172     getContentPane().setLayout(layout);
173     layout.setHorizontalGroup(
174         layout.createParallelGroup(javax.swing.GroupLayout.
175             Alignment.LEADING)
176             .addGap(0, 400, Short.MAX_VALUE)
177     );
178     layout.setVerticalGroup(
179         layout.createParallelGroup(javax.swing.GroupLayout.
180             Alignment.LEADING)
181             .addGap(0, 300, Short.MAX_VALUE)
182     );
183     pack();
184 } // </editor-fold> //GEN-END: initComponents
185
186 // Variables declaration - do not modify //GEN-BEGIN: variables
187 // End of variables declaration //GEN-END: variables
188 }

```

### 2.5.19. VentanaSimuladorDesc.java

```

1 //SimAS / Simulador
2 //Ventana simulador Descendente
3
4 package es.uco.simas.simulador;
5
6 import es.uco.simas.util.gramatica.Gramatica;
7 import es.uco.simas.SimAS;
8 import es.uco.simas.util.gramatica.NoTerminal;
9 import es.uco.simas.util.gramatica.Produccion;
10 import java.util.ArrayList;
11
12 /**
13  * @author Vanesa
14  */
15 public class VentanaSimuladorDesc extends javax.swing.JFrame {
16
17     private PanelNuevaSimDescPaso1 paso1;
18     private PanelNuevaSimDescPaso2 paso2;
19     private PanelNuevaSimDescPaso3 paso3;
20     private PanelNuevaSimDescPaso4 paso4;
21     private PanelNuevaSimDescPaso5 paso5;
22
23     Gramatica gramatica;
24     ArrayList<NoTerminal> noTerminales = new ArrayList<>();
25     ArrayList<Produccion> pr = new ArrayList<>();
26

```

```
27 public VentanaSimuladorDesc(Gramatica gramatica) {
28     initComponents();
29     this.gramatica = gramatica;
30
31     this.setResizable(false);
32
33     PanelNuevaSimDescPaso1 paso1 = new PanelNuevaSimDescPaso1(this)
34         ;
35     this.paso1 = paso1;
36
37     PanelNuevaSimDescPaso3 paso3 = new PanelNuevaSimDescPaso3(this)
38         ;
39     this.paso3 = paso3;
40
41     PanelNuevaSimDescPaso4 paso4 = new PanelNuevaSimDescPaso4(this)
42         ;
43     this.paso4 = paso4;
44
45     this.getContentPane().removeAll();
46     this.setContentPane(this.paso1);
47     this.pack();
48     this.validate();
49     this.setTitle("Simulador Descendente. Paso 1 de 5");
50     this.setVisible(true);
51     this.setLocationRelativeTo(null);
52 }
53
54 public Gramatica getGramatica(){
55     return this.gramatica;
56 }
57
58 public void setGramatica(Gramatica gramatica){
59     this.gramatica = gramatica;
60 }
61
62 public ArrayList<NoTerminal> getNoTerminales() {
63     return noTerminales;
64 }
65
66 public void setNoTerminales(ArrayList<NoTerminal> noTerminales) {
67     this.noTerminales = noTerminales;
68 }
69
70 public ArrayList<Produccion> getPr() {
71     return pr;
72 }
73
74 public void setPr(ArrayList<Produccion> pr) {
75     this.pr = pr;
76 }
77
78 public void cambiarPaso(int paso) {
79     switch(paso){
```

```
77
78         case 1:{
79
80             this.setContentPane(this.paso1);
81             this.pack();
82             this.setVisible(true);
83             this.validate();
84             this.setTitle("Simulador Descendente. Paso 1 de 5");
85             break;
86
87         }
88         case 2: {
89             this.setTitle("Simulador Descendente. Paso 2 de 5");
90             PanelNuevaSimDescPaso2 paso2 = new
91                 PanelNuevaSimDescPaso2(this);
92             this.paso2 = paso2;
93             this.setContentPane(this.paso2);
94             this.pack();
95             this.setVisible(true);
96             this.validate();
97             this.gramatica.generarConjPrim();
98             this.gramatica.generarConjSig();
99             this.paso2.construirConjuntos(this.gramatica);
100             this.setTitle("Simulador Descendente. Paso 2 de 5");
101             break;
102
103         }
104         case 3:{
105             this.setContentPane(this.paso3);
106             this.pack();
107             this.setVisible(true);
108             this.validate();
109             this.paso3.ConstruirTPredictiva(this.gramatica);
110             this.setTitle("Simulador Descendente. Paso 3 de 5");
111             break;
112
113         }
114         case 4:{
115             this.setContentPane(this.paso4);
116             this.pack();
117             this.setVisible(true);
118             this.validate();
119             this.setTitle("Simulador Descendente. Paso 4 de 5");
120             break;
121
122         }
123         case 5:{
124             PanelNuevaSimDescPaso5 paso5 = new
125                 PanelNuevaSimDescPaso5((VentanaSimuladorDesc) this);
126             this.paso5 = paso5;
127             this.setContentPane(this.paso5);
128             this.pack();
129             this.setVisible(true);
```

```

128         this.validate();
129         this.setTitle("Simulador Descendente. Paso 5 de 5");
130         break;
131
132     }
133 }
134
135
136 public void finalizarAsistente(){
137     this.dispose();
138     SimAS simas = new SimAS();
139     simas.lanzarSimulador(1, this.gramatica, -1);
140 }
141
142 /**
143  * This method is called from within the constructor to initialize
144  * the form.
145  * WARNING: Do NOT modify this code. The content of this method is
146  * always
147  * regenerated by the Form Editor.
148  */
149 @SuppressWarnings("unchecked")
150 // <editor-fold defaultstate="collapsed" desc="Generated Code">
151 // GEN-BEGIN: initComponents
152 private void initComponents() {
153
154     setDefaultCloseOperation(javax.swing.WindowConstants.
155         EXIT_ON_CLOSE);
156
157     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
158         getContentPane());
159     getContentPane().setLayout(layout);
160     layout.setHorizontalGroup(
161         layout.createParallelGroup(javax.swing.GroupLayout.
162             Alignment.LEADING)
163         .addGroup(layout.createSequentialGroup()
164             .addGap(0, 400, Short.MAX_VALUE)
165             .addContainerGap())
166     );
167     layout.setVerticalGroup(
168         layout.createParallelGroup(javax.swing.GroupLayout.
169             Alignment.LEADING)
170         .addGroup(layout.createSequentialGroup()
171             .addGap(0, 300, Short.MAX_VALUE)
172             .addContainerGap())
173     );
174
175     pack();
176 }
177 // </editor-fold>
178 // GEN-END: initComponents
179
180 // Variables declaration - do not modify
181 // GEN-BEGIN: variables
182 // End of variables declaration
183 // GEN-END: variables
184 }

```

## 2.6. Paquete Gramática

### 2.6.1. Antecedente.java

```

1 //SimAS / Gramatica
2 //Antecedente
3
4 package es.uco.simas.util.gramatica;
5
6 /**
7  * @author vanesa
8  */
9 public class Antecedente extends javax.swing.JFrame {
10
11     NoTerminal simboloNT = new NoTerminal(null, null);
12
13     public Antecedente() {
14         initComponents();
15     }
16
17     public NoTerminal getSimboloNT() {
18         return simboloNT;
19     }
20
21     public void setSimboloNT(NoTerminal simboloNT) {
22         this.simboloNT = simboloNT;
23     }
24
25
26     /**
27      * This method is called from within the constructor to initialize
28      * the form.
29      * WARNING: Do NOT modify this code. The content of this method is
30      * always
31      * regenerated by the Form Editor.
32      */
33     @SuppressWarnings("unchecked")
34     // <editor-fold defaultstate="collapsed" desc="Generated Code">
35     GEN-BEGIN: initComponents
36     private void initComponents() {
37
38         setDefaultCloseOperation(javax.swing.WindowConstants.
39             EXIT_ON_CLOSE);
40
41         javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
42             getContentPane());
43         getContentPane().setLayout(layout);
44         layout.setHorizontalGroup(
45             layout.createParallelGroup(javax.swing.GroupLayout.
46                 Alignment.LEADING)
47                 .addGap(0, 400, Short.MAX_VALUE)

```

```

42         );
43         layout.setVerticalGroup(
44             layout.createParallelGroup(javax.swing.GroupLayout.
45                 Alignment.LEADING)
46                 .addGap(0, 300, Short.MAX_VALUE)
47         );
48         pack();
49     }// </editor-fold>//GEN-END: initComponents
50
51     // Variables declaration - do not modify//GEN-BEGIN: variables
52     // End of variables declaration//GEN-END: variables
53 }

```

### 2.6.2. Consecuente.java

```

1  //SimAS / Gramatica
2  //Consecuente
3
4  package es.uco.simas.util.gramatica;
5
6  import java.util.ArrayList;
7
8  /**
9   * @author vanesa
10  */
11 public class Consecuente extends javax.swing.JFrame {
12
13     ArrayList<Simbolo> conjSimbolos = new ArrayList<>();
14
15     public Consecuente() {
16         initComponents();
17     }
18
19     public ArrayList<Simbolo> getConjSimbolos() {
20         return conjSimbolos;
21     }
22
23     public void setConjSimbolos(ArrayList<Simbolo> conjSimbolos) {
24         this.conjSimbolos = conjSimbolos;
25     }
26
27     /**
28      * This method is called from within the constructor to initialize
29      * the form.
30      * WARNING: Do NOT modify this code. The content of this method is
31      * always
32      * regenerated by the Form Editor.
33      */
34     @SuppressWarnings("unchecked")

```

```

33 // <editor-fold defaultstate="collapsed" desc="Generated Code">//
    GEN-BEGIN: initComponents
34 private void initComponents() {
35
36     setDefaultCloseOperation(javax.swing.WindowConstants.
        EXIT_ON_CLOSE);
37
38     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
        getContentPane());
39     getContentPane().setLayout(layout);
40     layout.setHorizontalGroup(
41         layout.createParallelGroup(javax.swing.GroupLayout.
            Alignment.LEADING)
42         .addGap(0, 400, Short.MAX_VALUE)
43     );
44     layout.setVerticalGroup(
45         layout.createParallelGroup(javax.swing.GroupLayout.
            Alignment.LEADING)
46         .addGap(0, 300, Short.MAX_VALUE)
47     );
48
49     pack();
50 }// </editor-fold>//GEN-END: initComponents
51
52 // Variables declaration - do not modify//GEN-BEGIN: variables
53 // End of variables declaration//GEN-END: variables
54 }

```

### 2.6.3. Gramatica.java

```

1 //SimAS / Gramatica
2 //Gramatica
3
4 package es.uco.simas.util.gramatica;
5
6 import com.itextpdf.text.BadElementException;
7 import com.itextpdf.text.BaseColor;
8 import com.itextpdf.text.Chunk;
9 import com.itextpdf.text.DocumentException;
10 import com.itextpdf.text.Font;
11 import com.itextpdf.text.Image;
12 import com.itextpdf.text.PageSize;
13 import com.itextpdf.text.Paragraph;
14 import com.itextpdf.text.pdf.BaseFont;
15 import com.itextpdf.text.pdf.draw.LineSeparator;
16 import com.itextpdf.text.pdf.PdfWriter;
17 import es.uco.simas.editor.ColCanLR0;
18 import es.uco.simas.editor.ColCanLR1;
19 import es.uco.simas.editor.ColCanLALR;
20 import es.uco.simas.editor.Editor;

```

```

21 import es.uco.simas.editor.TablaLR;
22 import es.uco.simas.editor.TablaPredictiva;
23 import java.util.*;
24 import java.io.File;
25 import javax.swing.JFileChooser;
26 import javax.swing.filechooser.FileNameExtensionFilter;
27 import javax.swing.DefaultListModel;
28 import java.io.*;
29 import java.net.MalformedURLException;
30 import java.util.logging.Level;
31 import java.util.logging.Logger;
32 import javax.xml.parsers.DocumentBuilder;
33 import javax.xml.parsers.DocumentBuilderFactory;
34 import javax.xml.parsers.ParserConfigurationException;
35 import org.w3c.dom.Document;
36 import org.w3c.dom.Element;
37 import org.w3c.dom.Node;
38 import org.w3c.dom.NodeList;
39 import org.xml.sax.SAXException;
40
41 /**
42  * @author vanesa
43  */
44 public class Gramatica extends javax.swing.JFrame {
45
46     public String nombre;
47     public String descripcion;
48     public String simbInicial;
49     public int estado;
50     ArrayList<Terminal> terminales = new ArrayList<>();
51     ArrayList<NoTerminal> noTerminales = new ArrayList<>();
52     ArrayList<Produccion> pr = new ArrayList<>();
53     DefaultListModel noTerm = new DefaultListModel();
54     DefaultListModel term = new DefaultListModel();
55     DefaultListModel producciones = new DefaultListModel();
56     TablaPredictiva tpredictiva = new TablaPredictiva();
57     TablaLR tlr;
58     ColCanLR0 coleccionLR0;
59     ColCanLR1 coleccionLR1;
60     ColCanLALR coleccionLALR;
61
62
63     public Gramatica (String nombre, String descripcion){
64         this.nombre = nombre;
65         this.descripcion = descripcion;
66     }
67
68     public Gramatica() {
69         initComponents();
70     }
71
72     /**

```



```

73      * This method is called from within the constructor to initialize
74      the form.
75      * WARNING: Do NOT modify this code. The content of this method is
76      always
77      * regenerated by the Form Editor.
78      */
79      @SuppressWarnings("unchecked")
80      // <editor-fold defaultstate="collapsed" desc="Generated Code">
81      GEN-BEGIN: initComponents
82      private void initComponents() {
83
84          setDefaultCloseOperation(javax.swing.WindowConstants.
85              EXIT_ON_CLOSE);
86
87          javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
88              getContentPane());
89          getContentPane().setLayout(layout);
90          layout.setHorizontalGroup(
91              layout.createParallelGroup(javax.swing.GroupLayout.
92                  Alignment.LEADING)
93                  .addGap(0, 400, Short.MAX_VALUE)
94          );
95          layout.setVerticalGroup(
96              layout.createParallelGroup(javax.swing.GroupLayout.
97                  Alignment.LEADING)
98                  .addGap(0, 300, Short.MAX_VALUE)
99          );
100          pack();
101      }
102      </editor-fold>
103      GEN-END: initComponents
104
105      public String getNombre() {
106          return nombre;
107      }
108
109      public void setNombre(String nombre) {
110          this.nombre = nombre;
111      }
112
113      public String getDescripcion() {
114          return descripcion;
115      }
116
117      public void setDescripcion(String descripcion) {
118          this.descripcion = descripcion;
119      }
120
121      public int getEstado() {
122          return estado;
123      }
124
125      public void setEstado(int estado) {
126          this.estado = estado;
127      }
128

```

```

119     }
120
121     public void setVocabulario( DefaultListModel noTerm,
122         DefaultListModel term) {
123         this.noTerm.clear();
124         this.noTerminales.clear();
125         this.term.clear();
126         this.terminales.clear();
127         int i= 0;
128         if(noTerm!= null) {
129             i=0;
130             while(true) {
131                 if(i >= noTerm.size()) {
132                     break;
133                 }
134                 if(i <noTerm.size()) {
135                     NoTerminal noterminal = new NoTerminal((String)
136                         noTerm.get(i),(String)noTerm.get(i));
137                     this.noTerminales.add(noterminal);
138                     i = i+1;
139                 }
140             }
141         }
142         if(term != null) {
143             i=0;
144             while(true) {
145                 if(i >= term.size()) {
146                     break;
147                 }
148                 if(i <term.size()) {
149                     Terminal terminal = new Terminal((String)term.get(i)
150                         ), (String)term.get(i));
151                     this.terminales.add(terminal);
152                     i = i+1;
153                 }
154             }
155         }
156     }
157
158     public DefaultListModel getNoTerminales() {
159         return this.noTerm;
160     }
161
162     public ArrayList<Terminal> getTerm() {
163         return terminales;
164     }
165
166     public ArrayList<NoTerminal> getNoTerm() {
167         return noTerminales;
168     }
169
170     public void setNoTerminales(DefaultListModel noTerminales){
171         this.noTerminales.clear();
172         this.noTerm = noTerminales;
173     }

```

```

169
170         int i=0;
171         if(noTerminales !=null){
172             while(i < this.noTerm.size()){
173                 NoTerminal nt = new NoTerminal(this.noTerm.getElementAt
                    (i).toString(),this.noTerm.getElementAt(i).toString
                    ());
174                 this.noTerminales.add(nt);
175                 i++;
176             }
177         }
178     }
179
180     public DefaultListModel getTerminales(){
181         return term;
182     }
183
184     public void setTerminales(DefaultListModel te){
185         this.terminales.clear();
186         this.term.clear();
187         int i=0;
188
189         if(te != null){
190             this.term = te;
191             if(terminales !=null){
192                 while(i < this.term.size()){
193                     Terminal t = new Terminal(this.term.getElementAt(i).
                        toString(),this.term.getElementAt(i).toString())
                        ;
194                     this.terminales.add(t);
195                     i++;
196                 }
197             }
198         }
199     }
200
201     public void setTerminales(ArrayList<Terminal> terminales) {
202         this.terminales = terminales;
203     }
204
205     public DefaultListModel getProducciones() {
206         return producciones;
207     }
208
209     public ArrayList<Produccion> getPr() {
210         return pr;
211     }
212
213     public void setPr(ArrayList<Produccion> pr) {
214         this.pr = pr;
215     }
216
217     public void setProducciones(DefaultListModel produc){

```

```

218
219     this.producciones = produc;
220     this.pr = new ArrayList();
221
222     int i,j= 0;
223     if(produc!= null) {
224         i=0;
225
226         while(i < produc.size()){
227
228             String valor = produc.elementAt(i).toString();
229             String antec = "";
230             String [] separado;
231             separado = valor.split(" ");
232
233             Antecedente antecedente = new Antecedente();
234             ArrayList<Simbolo> consecuente = new ArrayList<>();
235             antec = separado[0];
236
237
238             j=2;
239             while(j < separado.length){
240
241                 Simbolo simb = new Simbolo(separado[j],separado[j])
242                     ;
243
244                 consecuente.add(simb);
245                 j++;
246             }
247             NoTerminal nt = new NoTerminal(null,null);
248             nt.setValor(antec);
249             nt.setNombre(antec);
250             antecedente.setSimboloNT(nt);
251             Produccion produccion = new Produccion();
252             produccion.setAntec(antecedente);
253             produccion.setConsec(consecuente);
254             this.pr.add(produccion);
255
256             i ++;
257         }
258     }
259
260     public void setSimbInicial(String simInicial){
261         this.simbInicial = simInicial;
262     }
263
264     public String getSimbInicial(){
265         return this.simbInicial;
266     }
267
268     public void selecSimboloInicial(String simInicial){
269

```

```

270     int i= 0;
271     if (this.noTerminales!= null) {
272         i=0;
273         while(true) {
274             if(i >= this.noTerminales.size()) {
275                 break;
276             }
277             if(i <this.noTerminales.size()) {
278                 if(noTerminales.get(i).toString().equals(simInicial
279                     )){
280                     this.noTerminales.get(i).setSimboloInicial(true
281                         );
282                     break;
283                 }else{
284                     i = i+1;
285                 }
286             }
287         }
288     }
289
290     public int guardarGramatica( ) {
291         FileNameExtensionFilter filtro = new FileNameExtensionFilter("
292             Archivos de XML", "xml");
293         JFileChooser fileChooserGuardar = new JFileChooser();
294         fileChooserGuardar.setFileFilter(filtro);
295         fileChooserGuardar.setDialogTitle("Guardar");
296         String documentoXml= "";
297         FileWriter fstream= null;
298         BufferedWriter out= null;
299         String nombreFichero= null;
300         String causasError= null;
301         String codigoError= null;
302         JFileChooser selector= null;
303         String mensajeError= null;
304
305         int seleccion = fileChooserGuardar.showSaveDialog(null);
306
307         if (seleccion == JFileChooser.APPROVE_OPTION) {
308             File file = fileChooserGuardar.getSelectedFile();
309             StringBuilder doc = new StringBuilder();
310             documentoXml =doc.append(documentoXml).append("<?xml
311                 version=\"1.0\" encoding=\"UTF-8\"?>\n").toString();
312             StringBuilder doc2 = new StringBuilder();
313             documentoXml =doc2.append(documentoXml).append("<?xml-
314                 stylesheet type=\"text/xsl\" href=\"gramatica.xsl\"?>\n

```

```

315 documentoXml =nombre.append(documentoXml).append("\t<name>"
    ).append(this.getNombre()).append("</name>\n").toString
    ();
316 StringBuilder desc = new StringBuilder();
317 documentoXml =desc.append(documentoXml).append("\t<
    description>").append(this.getDescripcion()).append("</
    description>\n").toString();
318 StringBuilder nTerm = new StringBuilder();
319 documentoXml =nTerm.append(documentoXml).append("\t<non-
    terminal-symbols>\n").toString();
320 int i=0;
321 while(true) {
322     if(i >= this.noTerminales.size()) {
323         break;
324     }
325     if(i < this.noTerminales.size()) {
326         StringBuilder JdecGenerated162 = new StringBuilder
            ();
327         documentoXml =JdecGenerated162.append(documentoXml)
            .append("\t\t<non-terminal>\n\t\t\t\t<value>").
            append(this.getNoTerminales().get(i)).append("
            </value>\n\t\t\t</non-terminal>\n").toString();
328         i = i+1;
329         continue ;
330     }
331 }
332 }
333 StringBuilder JdecGenerated210 = new StringBuilder();
334 documentoXml =JdecGenerated210.append(documentoXml).append(
    "\t</non-terminal-symbols>\n").toString();
335 StringBuilder JdecGenerated230 = new StringBuilder();
336 documentoXml =JdecGenerated230.append(documentoXml).append(
    "\t<terminal-symbols>\n").toString();
337 i=0;
338 while(true) {
339     if(i >= this.terminales.size()) {
340         break;
341     }
342     if(i < this.terminales.size()) {
343         StringBuilder JdecGenerated263 = new StringBuilder();
344         documentoXml =JdecGenerated263.append(documentoXml).
            append("\t\t<terminal>\n\t\t\t\t\t\t<value>").append(
            this.getTerminales().get(i)).append("</value>\n\t\t
            \t</terminal>\n").toString();
345         i = i+1;
346         continue ;
347     }
348 }
349 }
350 StringBuilder JdecGenerated311 = new StringBuilder();
351 documentoXml =JdecGenerated311.append(documentoXml).append(
    "\t</terminal-symbols>\n").toString();
352 StringBuilder JdecGenerated331 = new StringBuilder();

```

```

353 documentoXml =JdecGenerated331.append(documentoXml).append(
    "\t<init-symbol>").append(this.getSimbInicial()).append(
    "</init-symbol>\n").toString();
354 StringBuilder JdecGenerated366 = new StringBuilder();
355 documentoXml =JdecGenerated366.append(documentoXml).append(
    "\t<rule-set>\n").toString();
356 i=0;
357 if(this.producciones != null){
358     while(true) {
359         if(i >= this.producciones.size()){
360             break;
361         }
362         if(i < this.producciones.size()) {
363             StringBuilder JdecGenerated399 = new
                StringBuilder();
364             documentoXml =JdecGenerated399.append(
                documentoXml).append("\t\t<rule>\n\t\t\t\t<
                value>").append(this.getProducciones().get(
                i)).append("</value>\n\t\t\t</rule>\n").
                toString();
365             i = i+1;
366         }
367     }
368 }
369 StringBuilder JdecGenerated859 = new StringBuilder();
370 documentoXml =JdecGenerated859.append(documentoXml).append(
    "\t</rule-set>\n").toString();
371 StringBuilder JdecGenerated879 = new StringBuilder();
372 documentoXml =JdecGenerated879.append(documentoXml).append(
    "</grammar>\n").toString();
373
374 try {
375     nombreFichero=fileChooserGuardar.getSelectedFile().
        toString();
376     if(!nombreFichero.contains(".xml")) {
377         StringBuilder JdecGenerated998 = new StringBuilder
            ();
378         nombreFichero =JdecGenerated998.append(
            nombreFichero).append(".xml").toString();
379     }
380
381     FileWriter JdecGenerated1020 = new FileWriter(
        nombreFichero);
382     fstream=JdecGenerated1020;
383     BufferedWriter JdecGenerated1031 = new BufferedWriter(
        fstream);
384     out=JdecGenerated1031;
385     out.write(documentoXml);
386     out.close();
387     return 1;
388 } catch(IOException e) {
389     codigoError="E-8";
390

```





```

424     DocumentBuilder db= null;
425     DocumentBuilderFactory dbf= null;
426     Element elemento= null;
427     Element elementoAntecedente= null;
428     Element elementoConsecuente= null;
429     Element elementoTipo= null;
430     Element elementoValor= null;
431     File file= null;
432     NoTerminal noTerminal= null;
433     Node nodo= null;
434     Node nodoAntecedente= null;
435     Node nodoConsecuente= null;
436     Node nodoTipo= null;
437     Node nodoValor= null;
438     NodeList description= null;
439     NodeList initSymbol= null;
440     NodeList name= null;
441     NodeList nodeNonTerminalSymbols= null;
442     NodeList nodeRules= null;
443     NodeList nodeTerminalSymbols= null;
444     NodeList valueNode= null;
445     NodeList valueNodeAntecedente= null;
446     NodeList valueNodeConsecuente= null;
447     NodeList valueNodoTipo= null;
448     NodeList valueNodoValor= null;
449     Produccion produccion= null;
450     String causasError= null;
451     String codigoError= null;
452     String descripcion= null;
453     String mensajeError= null;
454     String nombre= null;
455     String simboloInicial= null;
456     Terminal terminal= null;
457     DefaultListModel NT = new DefaultListModel();
458     DefaultListModel term = new DefaultListModel();
459     DefaultListModel prod = new DefaultListModel();
460
461     int i= 0;
462     int j= 0;
463     try {
464         if(nombreFichero != null) {
465             File JdecGenerated2 = new File(nombreFichero);
466             file=JdecGenerated2;
467             dbf=DocumentBuilderFactory.newInstance();
468             db=dbf.newDocumentBuilder();
469             doc=db.parse(file);
470             doc.getDocumentElement().normalize();
471             name=doc.getDocumentElement().getElementsByTagName("
                name");
472             nombre=name.item(0).getChildNodes().item(0).
                getNodeValue().toString();
473             description=doc.getDocumentElement().
                getElementsByTagName("description");

```

```

474      descripcion=description.item(0).getChildNodes().item(0)
475      .getNodeValue().toString();
476      Gramatica JdecGenerated129 = new Gramatica(nombre,
477      descripcion);
478      gramatica=JdecGenerated129;
479      nodeNonTerminalSymbols=doc.getElementsByTagName("non-
480      terminal");
481      i=0;
482      while(true) {
483          if(i >= nodeNonTerminalSymbols.getLength()) {
484              gramatica.setNoTerminales(NT);
485              break;
486          }
487          if(i < nodeNonTerminalSymbols.getLength()) {
488              nodo=nodeNonTerminalSymbols.item(i);
489              elemento=(Element)nodo;
490              valueNode=elemento.getElementsByTagName("value"
491              );
492              NT.addElement(valueNode.item(0).getChildNodes()
493              .item(0).getNodeValue());
494              i = i+1;
495              continue ;
496          }
497      }
498      nodeTerminalSymbols=doc.getElementsByTagName("terminal"
499      );
500      i=0;
501      while(true) {
502          if(i >= nodeTerminalSymbols.getLength()) {
503              gramatica.setTerminales(term);
504              break;
505          }
506          if(i < nodeTerminalSymbols.getLength()){
507              nodo=nodeTerminalSymbols.item(i);
508              elemento=(Element)nodo;
509              valueNode=elemento.getElementsByTagName("value"
510              );
511              term.addElement(valueNode.item(0).getChildNodes()
512              .item(0).getNodeValue());
513              i = i+1;
514              continue ;
515          }
516      }
517      this.setVocabulario(NT, term);
518      initSymbol=doc.getDocumentElement().
519      getElementsByTagName("init-symbol");
520      simboloInicial=initSymbol.item(0).getChildNodes().item
521      (0).getNodeValue().toString();
522      gramatica.setSimbInicial(simboloInicial);
523      nodeRules=doc.getElementsByTagName("rule");

```

```

517         i=0;
518         while(true) {
519             if(i >= nodeRules.getLength()) {
520                 break;
521             }
522             if(i < nodeRules.getLength()) {
523                 nodo=nodeRules.item(i);
524                 elemento=(Element)nodo;
525                 valueNode=elemento.getElementsByTagName("value");
526                 prod.addElement(valueNode.item(0).getChildNodes().item(0).getNodeValue());
527 
528                 i = i+1;
529             }
530         }
531         gramatica.setProducciones(prod);
532 
533         return gramatica;
534     }
535 
536 } catch(IOException e) {
537     codigoError="E-6";
538     mensajeError="Error de formato al cargar el fichero de gram\
539 u00eltica.";
540     StringBuilder JdecGenerated713 = new StringBuilder();
    causasError=JdecGenerated713.append("<br>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<b>Causas del error</b><br><blockquote><ul><li>El\nformato de la gram\u00eltica no es correcto.\n\u00bfHas\nmodificado el fichero manualmente?<br><br></li><li>Es\nposible que el fichero est\u00e9 corrupto, debido a un\nfallo del S.O. o del sistema de archivos.</li></ul></\nblockquote>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<b>Soluciones</b><\nblockquote><ul><li>Crea la gram\u00eltica de nuevo a\npartir de un informe de gram\u00eltica o intenta\nrecuperarla de otro fichero. Este error es cr\u00edtico y\nno se puede <b>recuperar</b> la gram\u00eltica del\nfichero da\u00f1ado.<br><br></li><li>Podr\u00edas\nintentar seguir la <b>excepci\u00f3n de java</b> e\nintentar <b>reconstruir</b> el fichero (no es\nrecomendable)</li></ul></blockquote>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<b>Excepci\u00f3n JAVA</b><blockquote><b>\nSAXParseException</b>:</b> ").append(e.getLocalizedMessage()).\ntoString().replaceAll("/","","\u2215").replaceAll("<",">").\nreplaceAll(">","").append("<br></blockquote>").toString()\n);\n}\ncatch (ParserConfigurationException ex) {\n    Logger.getLogger(Gramatica.class.getName()).log(Level.\n        SEVERE, null, ex);\n}\ncatch (SAXException ex) {\n    Logger.getLogger(Gramatica.class.getName()).log(Level.\n        SEVERE, null, ex);\n}

```

```

546     }
547
548     return null;
549 }
550
551 public ArrayList<String> validarGramatica( ) {
552
553     ArrayList<String> mensajesError = new ArrayList();
554     ArrayList<Simbolo> conjSimbolos= new ArrayList();
555     Antecedente antec = new Antecedente();
556
557     Iterator i= null;
558     Antecedente antecedente= null;
559     Produccion produccion= new Produccion();
560     Simbolo simbolo= null;
561     NoTerminal nt = null;
562     Terminal t = null;
563     int encontrado= 0;
564     int errorIndex= 0;
565     int simbolosNoTerminalesEncontrados= 0;
566     int simbolosTerminalesEncontrados= 0;
567     errorIndex=1;
568     this.setEstado(1);
569
570     if(this.producciones==null) {
571         this.setEstado(-1);
572         StringBuilder mensaje = new StringBuilder();
573         mensajesError.add(mensaje.append("<b>").append(errorIndex
            ++).append("</b>").append(" . <font color=\"red\"><i>No
            existen producciones</i>.</font><br> La gramática no
            contiene ninguna <b>produccion</b>. Debería contener
            al menos una para poder ser válida.<br></b>").
            toString());
574     }
575     if(this.terminales.isEmpty() == true) {
576         this.setEstado(-1);
577         StringBuilder mensaje = new StringBuilder();
578         mensajesError.add(mensaje.append("<b>").append(errorIndex
            ++).append("</b>").append(" . <font color=\"red\"><i>No
            existen símbolos terminales</i></font>.<br> La
            gramática no contiene ningún <b>símbolo terminal</b>
            >. Debería contener al menos uno para poder ser
            válida.<br></b>").toString());
579     }
580     if(this.noTerminales.isEmpty() == true){
581         this.setEstado(-1);
582         StringBuilder mensaje = new StringBuilder();
583         mensajesError.add(mensaje.append("<b>").append(errorIndex
            ++).append("</b>").append(" . <font color=\"red\"><i>No
            existen símbolos no terminales</i></font>.<br> La
            gramática no contiene ningún símbolo <b>no terminal
            </b>. Debería contener al menos uno para poder ser

```

```

        vÃ¡lida.<br></br>").toString());
584
585     }
586     if (this.simbInicial == null) {
587         this.setEstado(-1);
588         StringBuilder mensaje = new StringBuilder();
589         mensajesError.add(mensaje.append("<b>").append(errorIndex
            ++).append("</b>").append(" . <font color=\"red\"><i>
                SÃmbolo inicial no asignado</i></font>.<br> La
                gramÃtica no tiene asignado el sÃmbolo <b>inicial</b>
                >.<br></br>").toString());
590     }
591
592     //Simbolo terminal no aparece en el consecuente de ninguna
        produccion
593     int j=0;
594     int k=0;
595     int l=0;
596
597     while(j< this.terminales.size()){
598         t = this.terminales.get(j);
599         encontrado = 0;
600         k=0;
601         while(k < this.pr.size()){
602
603             conjSimbolos = pr.get(k).getConsec();
604             l=0;
605             while(l< conjSimbolos.size()){
606                 if (conjSimbolos.get(l).getValor().equals(t.getValor
                    ())) {
607                     encontrado = 1;
608                 }
609                 l++;
610             }
611             k++;
612         }
613         if (encontrado == 0) {
614             this.setEstado(-1);
615             StringBuilder str = new StringBuilder();
616             mensajesError.add(str.append("<b>").append(errorIndex
                ++).append("</b>").append(" . <font color=\"red\"><i>
                    >Simbolo terminal no usado</i></font>.<br> El
                    sÃmbolo terminal <b>").append(t.getNombre()).
                    append("</b> no aparece en ningÃn <b>consecuente</b>
                    <b> de ninguna produccion.<br></br>").toString());
617         }
618         j++;
619     }
620 }
621
622 //Simbolo no terminal no aparece en el consecuente de ninguna
        produccion
623 j=0;

```

```

624
625     while(j< this.noTerminales.size()){
626         nt = this.noTerminales.get(j);
627         encontrado = 0;
628         k=0;
629         while(k < this.pr.size()){
630
631             conjSimbolos = pr.get(k).getConsec();
632             l=0;
633             while(l< conjSimbolos.size()){
634                 if(nt.getValor().equals(this.getSimbInicial())){
635                     encontrado = 1;
636                     l++;
637                 }else{
638                     if(conjSimbolos.get(l).getValor().equals(nt.
639                         getValor())){
640                         encontrado = 1;
641                         l=conjSimbolos.size();
642                     } else
643                         l++;
644                 }
645             }
646             k++;
647         }
648         if(encontrado == 0){
649             this.setEstado(-1);
650             StringBuilder str = new StringBuilder();
651             mensajesError.add(str.append("<b>").append(errorIndex
652                 ++).append("</b>").append(" . <font color=\"red\"><i
653                 >Simbolo no terminal no usado</i></font>.<br> El
654                 sÃmbolo no terminal <b>").append(nt.getNombre()).
655                 append("</b> no aparece en ningÃn <b>consecuente</
656                 b> de ninguna produccion.<br></br>").toString());
657         }
658         j++;
659     }
660
661     //Simbolo no terminal no aparece en el antecedente de ninguna
662     produccion
663     j=0;
664
665     while(j< this.noTerminales.size()){
666         nt = this.noTerminales.get(j);
667         encontrado = 0;
668         k=0;
669         while(k < this.pr.size()){
670             produccion = this.pr.get(k);
671             antec = pr.get(k).getAntec();
672
673             if(nt.getValor().equals(antec.getSimboloNT().getValor()
674                 )){
675                 encontrado = 1;

```

```

669         }
670         k++;
671     }
672     if(encontrado==0) {
673         this.setEstado(-1);
674         StringBuilder str = new StringBuilder();
675         mensajesError.add(str.append("<b>").append(errorIndex
        ++).append("</b>").append(" . <font color=\"red\"><i
        >Simbolo no terminal no usado</i></font>.<br> El
        símbolo no terminal <b>").append(nt.getNombre()).
        append("</b> no aparece en el <b>antecedente</b> de
        ninguna produccion.<br></br>").toString());
676     }
677     j++;
678 }
679
680 //Simbolo no terminal del antecedente no esta en el conjunto de
        simbolos no terminales de la gramatica
681 j=0;
682 while(j < this.pr.size()){
683
684     nt = new NoTerminal(this.pr.get(j).getAntec().getSimboloNT
        ().getValor(), this.pr.get(j).getAntec().getSimboloNT()
        .getValor());
685
686     encontrado = 0;
687     k=0;
688     while(k< this.noTerminales.size()){
689
690         if(nt.getValor().equals(this.noTerminales.get(k).
        getValor())){
691             encontrado = 1;
692         }
693         k++;
694     }
695     if(encontrado == 0){
696         this.setEstado(-1);
697         StringBuilder str = new StringBuilder();
698         mensajesError.add(str.append("<b>").append(errorIndex
        ++).append("</b>").append(" . <font color=\"red\"><i
        >Simbolo no terminal inexistente</i></font><br>. El
        símbolo no terminal <b>").append(this.pr.get(j).
        getAntec().getSimboloNT().getNombre()).append("</b
        >, que es el antecedente de la produccion <b>").
        append("</b>, no está contenido en el conjunto de
        símbolos <b>no terminales</b> de la gramática.<br
        ></br>").toString());
699     }
700     j++;
701 }
702
703 //Simbolo no terminal del consecuente no está en el conjunto
        de Simbolos no terminales

```

```

704         j=0;
705
706         while(j < this.pr.size()){
707             conjSimbolos = pr.get(j).getConsec();
708
709             encontrado = 0;
710             k=0;
711
712             while(k< conjSimbolos.size()){
713                 encontrado = 0;
714                 simbolo = conjSimbolos.get(k);
715
716                 if(simbolo.getValor().equals("\u03b5")){
717                     encontrado = 1;
718                 }else{
719                     l=0;
720                     while(l < this.noTerminales.size()){
721                         nt = this.noTerminales.get(l);
722
723                         if(nt.getValor().equals(simbolo.getValor())){
724                             encontrado = 1;
725                         }
726                         l++;
727                     }
728                     int m=0;
729                     if(encontrado ==0){
730                         while(m < this.terminales.size()){
731                             t = this.terminales.get(m);
732
733                             if(t.getValor().equals(simbolo.getValor()))
734                                 {
735                                     encontrado = 1;
736                                 }
737                             m++;
738                         }
739                     }
740                 }
741             }
742             k++;
743
744             if(encontrado == 0){
745                 this.setEstado(-1);
746                 StringBuilder str = new StringBuilder();
747                 mensajesError.add(str.append("<b>").append(
                    errorIndex++).append("</b>").append(". <font
                    color=\"red\"><i>Consecuente erroneo</i></font
                    >.<br> El sÃmbolo <b>").append(simbolo.
                    getNombre()).append("</b> del <b>consecuente</b>
                    > de la produccion, no pertenece al conjunto de
                    sÃmbolos declarado.<br></br>").toString());
748             }
749         }

```



```

750         j++;
751     }
752
753     return mensajesError;
754 }
755
756 public Boolean generarInforme(String fichero) throws
    DocumentException{
757
758     if (this.estado==1){
759         try {
760             String font = "fonts/arial.ttf";
761             com.itextpdf.text.Document document = new com.itextpdf.
                text.Document(PageSize.LETTER , 45, 45, 54, 45);
762             Image imagen = Image.getInstance("./src/es/uco/simas/
                resources/logo2Antes.png");
763             imagen.setAlignment(com.itextpdf.text.Element.
                ALIGN.CENTER);
764             imagen.scalePercent(40);
765
766             LineSeparator ls = new LineSeparator();
767             BaseFont bf;
768             bf = BaseFont.createFont(font, BaseFont.IDENTITY_H,
                BaseFont.EMBEDDED);
769
770             Font titulo = new Font(bf, 21, Font.BOLD);
771             Font font2 = new Font(bf, 15, Font.BOLD);
772             Font font3 = new Font(bf, 12);
773             BaseColor claro = new BaseColor(63,171,160);
774
775             titulo.setColor(33, 77, 72);
776             font2.setColor(BaseColor.BLACK);
777
778             ls.setLineWidth(1);
779             ls.setLineColor(claro);
780
781             Paragraph parrafo = new Paragraph(" INFORME DE LA
                GRAMATICA ", titulo);
782             parrafo.setAlignment(com.itextpdf.text.Element.
                ALIGN.CENTER);
783             Paragraph parrafo1 = new Paragraph("\n Nombre de la
                gramática: ", font2);
784             parrafo1.add(new Paragraph(" "+this.nombre+"\n",
                font3));
785             Paragraph parrafo2 = new Paragraph("\n Descripcion de
                la gramática: ", font2);
786             parrafo2.add(new Paragraph(" "+this.descripcion+
                "\n", font3));
787             Paragraph parrafo3 = new Paragraph("\n Símbolos
                terminales: ", font2);
788             DefaultListModel term = this.term;
789             int i=0;
790             while(i<term.getSize()){

```

```

791
792         parrafo3.add(new Paragraph("      "+term.
793             getElementAt(i).toString(), font3));
794         i++;
795     }
796     Paragraph parrafo4 = new Paragraph("\n S mbolos no
797     terminales: ", font2);
798     DefaultListModel nterm = this.noTerm;
799     i=0;
800     while(i<nterm.getSize()){
801         parrafo4.add(new Paragraph("      "+nterm.
802             getElementAt(i).toString(), font3));
803         i++;
804     }
805     Paragraph parrafo5 = new Paragraph("\n S mbolo
806     inicial de la gram tica: ", font2);
807     parrafo5.add(new Paragraph("      "+this.simbInicial,
808         font3));
809     Paragraph parrafo6 = new Paragraph("\n Producciones de
810     la gram tica: ", font2);
811
812     DefaultListModel produc = this.producciones;
813     DefaultListModel produc2 = new DefaultListModel();
814     i =0;
815     Object obj;
816     obj = "P {";
817     produc2.addElement(obj);
818     while (i < produc.size()){
819         obj = "      "+(i+1)+"      "+produc.getElementAt(i)
820             ;
821         produc2.addElement(obj);
822         i++;
823     }
824     obj ="}";
825     produc2.addElement(obj);
826
827     i=0;
828     while(i<produc2.getSize()){
829         parrafo6.add(new Paragraph("      "+produc2.
830             getElementAt(i).toString(), font3));
831         i++;
832     }
833
834     try {
835         PdfWriter.getInstance(document, new
836             FileOutputStream(fichero));
837     } catch (DocumentException | FileNotFoundException ex) {
838         Logger.getLogger(Editor.class.getName()).log(Level.
839             SEVERE, null, ex);
840     }

```

```
834
835         document.open();
836     try {
837
838         document.add(imagen);
839         document.add(parrafo);
840         document.add(new Chunk(ls));
841         document.add(parrafo1);
842         document.add(parrafo2);
843         document.add(parrafo3);
844         document.add(parrafo4);
845         document.add(parrafo5);
846         document.add(parrafo6);
847
848     } catch (DocumentException ex) {
849         Logger.getLogger(Editor.class.getName()).log(Level.
            SEVERE, null, ex);
850     }
851
852     document.close();
853 } catch (BadElementException ex) {
854     Logger.getLogger(Editor.class.getName()).log(Level.
        SEVERE, null, ex);
855 } catch (MalformedURLException ex) {
856     Logger.getLogger(Editor.class.getName()).log(Level.
        SEVERE, null, ex);
857 } catch (IOException ex) {
858     Logger.getLogger(Editor.class.getName()).log(Level.
        SEVERE, null, ex);
859 }
860 } else {
861     return false;
862 }
863 }
864
865 return true;
866 }
867
868 public boolean isNoTerminal (String nombre){
869     int i=0;
870     int encontrado = 0;
871     ArrayList<NoTerminal> noTerminales = this.getNoTerm();
872
873     while(i < noTerminales.size()){
874         if(noTerminales.get(i).getNombre().equals(nombre)){
875             encontrado = 1;
876             break;
877         } else {
878             i++;
879         }
880     }
881     if(encontrado == 1)
882         return true;
```

```

883         else
884             return false;
885     }
886
887     public boolean isTerminal (String nombre){
888         int i=0;
889         int encontrado = 0;
890         ArrayList<Terminal> terminales = this.getTerm();
891
892         while(i < terminales.size()){
893             if(terminales.get(i).getNombre().equals(nombre)){
894                 encontrado = 1;
895                 break;
896             }else{
897                 i++;
898             }
899         }
900         if(encontrado == 1)
901             return true;
902         else
903             return false;
904     }
905
906     public void generarConjPrim(){
907         int i=0;
908         int j=0;
909         int k=0;
910         int encontrado=0;
911         Iterator it;
912         ArrayList<Terminal> terminales = new ArrayList();
913         ArrayList<Produccion> producciones = new ArrayList();
914         ArrayList<Terminal> conjPrim = new ArrayList();
915         Map<String, String> despues = new HashMap<String, String>();
916         Terminal term;
917         NoTerminal nterm;
918         Produccion pr;
919         Simbolo primero;
920         terminales = this.getTerm();
921         producciones = this.getPr();
922
923         while(i < producciones.size()){
924             pr = producciones.get(i);
925             nterm = pr.getAntec().getSimboloNT(); //Antecedente
926             j=0;
927             while(j < this.noTerminales.size()){
928                 if(this.noTerminales.get(j).getNombre().equals(nterm.
929                     getNombre())){
930                     primero = pr.getConsec().get(0); //Primer simbolo
931                     //del consecuente
932                     conjPrim = this.noTerminales.get(j).getPrimeros();
933
934                     if(primero.getNombre().equals("\u03b5")){ // El
935                         //primer simbolo del consecuente es epsilon

```

```

933         k=0;
934         encontrado=0;
935         while(k < conjPrim.size()){
936             if(conjPrim.get(k).getNombre().equals("\u03b5")){
937                 encontrado = 1;
938             }
939             k++;
940         }
941         if(encontrado == 0)
942             conjPrim.add(new Terminal("\u03b5", "\u03b5"));
943     }else {
944         if(isTerminal(primer.getNombre())){// El
945             primer simbolo del consecuente es Terminal
946             k=0;
947             encontrado=0;
948             while(k < conjPrim.size()){
949                 if(primer.getNombre().equals(conjPrim.
950                     get(k).getNombre())){
951                     encontrado = 1;
952                 }
953                 k++;
954             }
955             if(encontrado == 0){
956                 conjPrim.add(new Terminal (primer.
957                     getNombre(), primer.getNombre()));
958             }
959         }else{ // El primer simbolo del consecuente es
960             no Terminal
961             if(!this.noTerminales.get(j).getNombre().
962                 equals(primer.getNombre())){
963                 despues.put(this.noTerminales.get(j).
964                     getNombre(), primer.getNombre());
965             }
966         }
967     }
968     }
969     }
970     // Si el primer simbolo del consecuente es un terminal se
971     resuelve
972     int x=0;
973     while(x < 2){
974         it = despues.keySet().iterator();
975         while(it.hasNext()){
976             String nterminal = (String) it.next();
977             String prim = despues.get(nterminal);

```

```

977         j=0;
978         while(j < this.noTerminales.size()){
979             if(this.noTerminales.get(j).getNombre().equals(prim
980                 )){
981                 terminales = this.noTerminales.get(j).
982                     getPrimeros();
983                 break;
984             }else
985                 j++;
986         }
987         j=0;
988         int indice = -1;
989         while(j < this.noTerminales.size()){
990             if(this.noTerminales.get(j).getNombre().equals(
991                 nterminal)){
992                 conjPrim = this.noTerminales.get(j).getPrimeros
993                     ();
994                 indice = j;
995                 break;
996             }else
997                 j++;
998         }
999         j=0;
1000         while(j < terminales.size()){
1001             encontrado = 0;
1002             k=0;
1003             while(k < conjPrim.size()){
1004                 if(conjPrim.get(k).getNombre().equals(
1005                     terminales.get(j).getNombre())){
1006                     encontrado = 1;
1007                     break;
1008                 }else
1009                     k++;
1010             }
1011             if(encontrado == 0){
1012                 conjPrim.add(new Terminal(terminales.get(j).
1013                     getNombre(), terminales.get(j).getNombre())
1014                     );
1015             }
1016             j++;
1017         }
1018         this.noTerminales.get(indice).setPrimeros(conjPrim);
1019     }
1020     x++;
1021 }
1022 }

1017 public void generarConjSig() {
1018     int i=0;
1019     int j=0;
1020     int k=0;
1021     int l=0;
1022     int m=0;

```

```

1023     int n=0;
1024     int o=0;
1025     int encontrado = 0;
1026     ArrayList<Terminal> conjSig = new ArrayList();
1027     ArrayList<Terminal> conjSig2 = new ArrayList();
1028     ArrayList<Terminal> conjPrim = new ArrayList();
1029     ArrayList<Produccion> producciones = this.getPr();
1030     ArrayList<Simbolo> consecuente = new ArrayList();
1031     Terminal terminal;
1032     NoTerminal antecedente;
1033
1034     while(i < this.noTerminales.size()){ // $ al simbolo inicial
1035         if(this.noTerminales.get(i).getNombre().equals(this.
            simbInicial)){
1036             conjSig.add(new Terminal("$", "$"));
1037             this.noTerminales.get(i).setSiguientes(conjSig);
1038             break;
1039         }else
1040             i++;
1041     }
1042     i=0;
1043     while(i < producciones.size()){
1044         consecuente = producciones.get(i).getConsec();
1045         j=1;
1046         while(j < consecuente.size()){
1047             //Si el simbolo posterior a un No terminal es un
            terminal forma parte del conjunto siguiente
1048             if(isNoTerminal(consecuente.get(j-1).getNombre()) &&
                isTerminal(consecuente.get(j).getNombre())){
1049                 k=0;
1050                 while(k < this.noTerminales.size()){
1051                     if(this.noTerminales.get(k).getNombre().equals(
                        consecuente.get(j-1).getNombre())){
1052                         conjSig = this.noTerminales.get(k).
                            getSiguientes();
1053
1054                         l=0;
1055                         encontrado = 0;
1056                         while(l < conjSig.size()){
1057                             if(conjSig.get(l).getNombre().equals(
                                    consecuente.get(j).getNombre())){
1058                                 encontrado = 1;
1059                                 break;
1060                             }else
1061                                 l++;
1062                         }
1063                         if(encontrado == 0){
1064                             conjSig.add(new Terminal (consecuente.
                                    get(j).getNombre(), consecuente.get(
                                        j).getNombre()));
1065                         }
1066                         this.noTerminales.get(k).setSiguientes(
                            conjSig);

```





```

1110         conjSig.add(new Terminal (conjPrim
1111             .get(1).getNombre(),conjPrim.
1112             get(1).getNombre()));
1113     }
1114     //Si el conjunto primero contiene la
1115     //palabra vacia:
1116     if (conjPrim.get(1).getNombre().equals(
1117         "\u03b5")){
1118         m=0;
1119         encontrado = 0;
1120         n=0;
1121         while(n < this.noTerminales.size()
1122             ){
1123             if (this.noTerminales.get(n).
1124                 getNombre().equals(
1125                     antecedente.getNombre())){
1126                 conjSig2 = this.
1127                     noTerminales.get(n).
1128                     getSiguientes();
1129                 break;
1130             }else
1131                 n++;
1132         }
1133         while(m < conjSig2.size()){
1134             n=0;
1135             encontrado =0;
1136             while(n < conjSig2.size()){
1137                 if (conjSig2.get(m).
1138                     getNombre().equals(
1139                         conjSig.get(n).
1140                         getNombre())){
1141                     encontrado = 1;
1142                     break;
1143                 }else
1144                     n++;
1145             }
1146             if(encontrado ==0){
1147                 conjSig.add(new Terminal(
1148                     conjSig2.get(m).
1149                     getNombre(), conjSig2.
1150                     get(m).getNombre()));
1151             }
1152             m++;
1153         }
1154         m=0;
1155         encontrado = 0;
1156         n=0;

```

```

1147 while(n < this.noTerminales.size()
1148 ) {
1149     if(this.noTerminales.get(n).
1150         getNombre().equals(
1151             consecuente.get(j).
1152             getNombre())) {
1153         conjSig = this.
1154             noTerminales.get(n).
1155             getSiguientes();
1156         break;
1157     } else
1158         n++;
1159 }
1160 while(m < conjSig2.size()) {
1161     encontrado=0;
1162     n=0;
1163     while(n < conjSig.size()) {
1164         if(conjSig2.get(m).
1165             getNombre().equals(
1166                 conjSig.get(n).
1167                 getNombre())) {
1168             encontrado = 1;
1169             break;
1170         } else
1171             n++;
1172     }
1173     if(encontrado ==0){
1174         conjSig.add(new Terminal(
1175             conjSig2.get(m).
1176             getNombre(), conjSig2.
1177             get(m).getNombre()));
1178     }
1179     m++;
1180 }
1181 m=0;
1182 encontrado = 0;
1183 n=0;
1184 while(n < this.noTerminales.size()
1185 ) {
1186     if(this.noTerminales.get(n).
1187         getNombre().equals(
1188             antecedente.getNombre())) {
1189         conjSig2 = this.
1190             noTerminales.get(n).
1191             getSiguientes();
1192         break;
1193     } else
1194         n++;
1195 }

```

```

1183
1184         n=0;
1185         while(n < this.noTerminales.size()
1186             ){
1187             if(this.noTerminales.get(n).
1188                 getNombre().equals(
1189                     consecuente.get(j-1).
1190                     getNombre())){
1191
1192                 conjSig = this.
1193                     noTerminales.get(n).
1194                     getSiguientes();
1195                 break;
1196             }else
1197                 n++;
1198         }
1199
1200         while(m < conjSig2.size()){
1201             encontrado=0;
1202             n=0;
1203             while(n < conjSig.size()){
1204                 if(conjSig2.get(m).
1205                     getNombre().equals(
1206                         conjSig.get(n).
1207                         getNombre())){
1208                     encontrado = 1;
1209                     break;
1210                 }else
1211                     n++;
1212             }
1213             if(encontrado ==0){
1214                 conjSig.add(new Terminal(
1215                     conjSig2.get(m).
1216                     getNombre(), conjSig2.
1217                     get(m).getNombre()));
1218             }
1219             m++;
1220         }
1221     }
1222     l++;
1223 }
1224
1225     }
1226     k++;
1227 }
1228
1229 if(isNoTerminal(consecuente.get(consecuente.size()-1).
1230     getNombre())){
1231     m=0;
1232     encontrado = 0;
1233     n=0;
1234     while(n < this.noTerminales.size()){

```

```

1223         if( this.noTerminales.get(n).getNombre().equals
1224             (antecedente.getNombre())){
1225             conjSig2 = this.noTerminales.get(n).
1226                 getSiguientes();
1227             break;
1228         }else
1229             n++;
1230     }
1231     n=0;
1232     while(n < this.noTerminales.size()){
1233         if( this.noTerminales.get(n).getNombre().equals
1234             (consecuente.get(consecuente.size()-1).
1235                 getNombre())){
1236             conjSig = this.noTerminales.get(n).
1237                 getSiguientes();
1238             break;
1239         }else
1240             n++;
1241     }
1242     while(m < conjSig2.size()){
1243         n=0;
1244         encontrado =0;
1245         while(n < conjSig.size()){
1246             if(conjSig2.get(m).getNombre().equals(
1247                 conjSig.get(n).getNombre())){
1248                 encontrado = 1;
1249                 break;
1250             }else
1251                 n++;
1252         }
1253         if(encontrado ==0){
1254             conjSig.add(new Terminal(conjSig2.get(m).
1255                 getNombre(), conjSig2.get(m).getNombre
1256                 ()));
1257         }
1258         m++;
1259     }
1260     }
1261     j++;
1262     }
1263     i++;
1264 }
1265 i=0;
1266 while(i < producciones.size()){
1267     consecuente = producciones.get(i).getConsec();
1268     antecedente = producciones.get(i).getAntec().getSimboloNT();
1269     j=0;
1270     while(j < consecuente.size()){

```

```

1267         if (isNoTerminal(consecuente.get(consecuente.size()-1).
1268             getNombre())){
1269             m=0;
1270             encontrado = 0;
1271             n=0;
1272             while(n < this.noTerminales.size()){
1273                 if (this.noTerminales.get(n).getNombre().equals
1274                     (antecedente.getNombre())){
1275                     conjSig2 = this.noTerminales.get(n).
1276                         getSiguientes();
1277                     break;
1278                 }else
1279                     n++;
1280             }
1281             n=0;
1282             while(n < this.noTerminales.size()){
1283                 if (this.noTerminales.get(n).getNombre().equals
1284                     (consecuente.get(consecuente.size()-1).
1285                         getNombre())){
1286                     conjSig = this.noTerminales.get(n).
1287                         getSiguientes();
1288                     break;
1289                 }else
1290                     n++;
1291             }
1292             while(m < conjSig2.size()){
1293                 n=0;
1294                 encontrado =0;
1295                 while(n < conjSig.size()){
1296                     if (conjSig2.get(m).getNombre().equals(
1297                         conjSig.get(n).getNombre())){
1298                         encontrado = 1;
1299                         break;
1300                     }else
1301                         n++;
1302                 }
1303                 if(encontrado ==0){
1304                     conjSig.add(new Terminal(conjSig2.get(m).
1305                         getNombre(), conjSig2.get(m).getNombre
1306                         ()));
1307                 }
1308                 m++;
1309             }
1310             j++;
1311         }
1312         i++;
1313     }

```

```

1311
1312 public void generarTPredictiva(){
1313     this.tpredictiva.construir(this);
1314 }
1315
1316 public TablaPredictiva getTPredictiva(){
1317
1318     return this.tpredictiva;
1319 }
1320
1321 public void setTPredictiva(TablaPredictiva tabla){
1322     this.tpredictiva = tabla;
1323 }
1324
1325 public void generarTLR(int i){
1326     this.tlr = new TablaLR(this);
1327     this.tlr.construir(i);
1328 }
1329
1330 public TablaLR getTlr() {
1331     return tlr;
1332 }
1333
1334 public void setTlr(TablaLR tlr) {
1335     this.tlr = tlr;
1336 }
1337
1338 public ColCanLR0 getColeccionLR0() {
1339     return this.coleccionLR0;
1340 }
1341
1342 public void setColeccionLR0(ColCanLR0 coleccionLR0) {
1343     this.coleccionLR0 = coleccionLR0;
1344 }
1345
1346 public ColCanLR1 getColeccionLR1() {
1347     return coleccionLR1;
1348 }
1349
1350 public void setColeccionLR1(ColCanLR1 coleccionLR1) {
1351     this.coleccionLR1 = coleccionLR1;
1352 }
1353
1354 public ColCanLALR getColeccionLALR() {
1355     return coleccionLALR;
1356 }
1357
1358 public void setColeccionLALR(ColCanLALR coleccionLALR) {
1359     this.coleccionLALR = coleccionLALR;
1360 }
1361
1362 // Variables declaration – do not modify//GEN-BEGIN:variables
1363 // End of variables declaration//GEN-END:variables

```

1364 }

#### 2.6.4. NoTerminal.java

```
1 //SimAS / Gramatica
2 //No Terminal
3
4 package es.uco.simas.util.gramatica;
5 import java.util.*;
6
7 /**
8  * @author vanesa
9  */
10 public class NoTerminal extends Simbolo {
11
12     boolean simboloInicial;
13     ArrayList<Terminal> primeros = new ArrayList<>();
14     ArrayList<Terminal> siguientes = new ArrayList<>();
15
16     public NoTerminal(String nombre, String valor) {
17         super(nombre, valor);
18         initComponents();
19     }
20
21     public boolean getSimboloInicial() {
22         return simboloInicial;
23     }
24
25     public void setSimboloInicial(boolean simboloInicial) {
26         this.simboloInicial = simboloInicial;
27     }
28
29     public ArrayList<Terminal> getPrimeros() {
30         return primeros;
31     }
32
33     public void setPrimeros(ArrayList<Terminal> primeros) {
34         this.primeros = primeros;
35     }
36
37     public ArrayList<Terminal> getSiguientes() {
38         return siguientes;
39     }
40
41     public void setSiguientes(ArrayList<Terminal> siguientes) {
42         this.siguientes = siguientes;
43     }
44
45     /**
```

```

46      * This method is called from within the constructor to initialize
    the form.
47      * WARNING: Do NOT modify this code. The content of this method is
    always
48      * regenerated by the Form Editor.
49      */
50      @SuppressWarnings("unchecked")
51      // <editor-fold defaultstate="collapsed" desc="Generated Code">//
    GEN-BEGIN:initComponents
52      private void initComponents() {
53
54          setDefaultCloseOperation(javax.swing.WindowConstants.
            EXIT_ON_CLOSE);
55
56          javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
            getContentPane());
57          getContentPane().setLayout(layout);
58          layout.setHorizontalGroup(
59              layout.createParallelGroup(javax.swing.GroupLayout.
                Alignment.LEADING)
60                  .addGap(0, 400, Short.MAX_VALUE)
61          );
62          layout.setVerticalGroup(
63              layout.createParallelGroup(javax.swing.GroupLayout.
                Alignment.LEADING)
64                  .addGap(0, 300, Short.MAX_VALUE)
65          );
66
67          pack();
68      }// </editor-fold>//GEN-END:initComponents
69
70      // Variables declaration - do not modify//GEN-BEGIN:variables
71      // End of variables declaration//GEN-END:variables
72  }

```

### 2.6.5. Produccion.java

```

1 //SimAS / Gramatica
2 //Produccion
3
4 package es.uco.simas.util.gramatica;
5
6 import java.util.ArrayList;
7
8 /**
9  * @author vanesa
10 */
11 public class Produccion extends javax.swing.JFrame {
12
13     Antecedente antec = new Antecedente();

```



```

14     ArrayList<Simbolo> consec = null;
15
16     public Produccion() {
17         initComponents();
18     }
19
20     public Antecedente getAntec() {
21         return antec;
22     }
23
24     public void setAntec(Antecedente antec) {
25         this.antec = antec;
26     }
27
28     public ArrayList<Simbolo> getConsec() {
29         return consec;
30     }
31
32     public void setConsec(ArrayList<Simbolo> consec) {
33         this.consec = consec;
34     }
35
36     /**
37      * This method is called from within the constructor to initialize
38      * the form.
39      * WARNING: Do NOT modify this code. The content of this method is
40      * always
41      * regenerated by the Form Editor.
42      */
43     @SuppressWarnings("unchecked")
44     // <editor-fold defaultstate="collapsed" desc="Generated Code">
45     // GEN-BEGIN: initComponents
46     private void initComponents() {
47
48         setDefaultCloseOperation(javax.swing.WindowConstants.
49             EXIT_ON_CLOSE);
50
51         javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
52             getContentPane());
53         getContentPane().setLayout(layout);
54         layout.setHorizontalGroup(
55             layout.createParallelGroup(javax.swing.GroupLayout.
56                 Alignment.LEADING)
57                 .addGap(0, 400, Short.MAX_VALUE)
58         );
59         layout.setVerticalGroup(
60             layout.createParallelGroup(javax.swing.GroupLayout.
61                 Alignment.LEADING)
62                 .addGap(0, 300, Short.MAX_VALUE)
63         );
64
65         pack();
66     }
67     // </editor-fold>
68     // GEN-END: initComponents

```

```

60
61 // Variables declaration – do not modify//GEN-BEGIN:variables
62 // End of variables declaration//GEN-END:variables
63 }

```

### 2.6.6. Simbolo.java

```

1 //SimAS / Gramatica
2 // Simbolo
3
4 package es.uco.simas.util.gramatica;
5
6 /**
7  * @author vanesa
8  */
9 public class Simbolo extends javax.swing.JFrame {
10
11     String nombre=null;
12     String valor=null;
13
14     public Simbolo(String nombre, String valor) {
15         this.nombre = nombre;
16         this.valor = valor;
17     }
18
19     public String getNombre() {
20         return nombre;
21     }
22
23     public void setNombre(String nombre) {
24         this.nombre = nombre;
25     }
26
27     public String getValor() {
28         return valor;
29     }
30
31     public void setValor(String valor) {
32         this.valor = valor;
33     }
34
35
36
37     /**
38      * This method is called from within the constructor to initialize
39      * the form.
40      * WARNING: Do NOT modify this code. The content of this method is
41      * always
42      * regenerated by the Form Editor.
43      */

```

```

42  @SuppressWarnings("unchecked")
43  // <editor-fold defaultstate="collapsed" desc="Generated Code">//
    GEN-BEGIN: initComponents
44  private void initComponents() {
45
46      setDefaultCloseOperation(javax.swing.WindowConstants.
        EXIT_ON_CLOSE);
47
48      javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
        getContentPane());
49      getContentPane().setLayout(layout);
50      layout.setHorizontalGroup(
51          layout.createParallelGroup(javax.swing.GroupLayout.
            Alignment.LEADING)
52              .addGap(0, 400, Short.MAX_VALUE)
53      );
54      layout.setVerticalGroup(
55          layout.createParallelGroup(javax.swing.GroupLayout.
            Alignment.LEADING)
56              .addGap(0, 300, Short.MAX_VALUE)
57      );
58
59      pack();
60  } // </editor-fold> // GEN-END: initComponents
61
62  /**
63   * @param args the command line arguments
64   */
65  public static void main(String args[]) {
66      /* Set the Nimbus look and feel */
67      //<editor-fold defaultstate="collapsed" desc=" Look and feel
        setting code (optional) ">
68      /* If Nimbus (introduced in Java SE 6) is not available, stay
        with the default look and feel.
69      * For details see http://download.oracle.com/javase/tutorial/
        uiswing/lookandfeel/plaf.html
70      */
71      try {
72          for (javax.swing.UIManager.LookAndFeelInfo info : javax.
            swing.UIManager.getInstalledLookAndFeels()) {
73              if ("Nimbus".equals(info.getName())) {
74                  javax.swing.UIManager.setLookAndFeel(info.
                    getClassName());
75                  break;
76              }
77          }
78      } catch (ClassNotFoundException ex) {
79          java.util.logging.Logger.getLogger(Simbolo.class.getName())
            .log(java.util.logging.Level.SEVERE, null, ex);
80      } catch (InstantiationException ex) {
81          java.util.logging.Logger.getLogger(Simbolo.class.getName())
            .log(java.util.logging.Level.SEVERE, null, ex);
82      } catch (IllegalAccessException ex) {

```

```

83         java.util.logging.Logger.getLogger(Simbolo.class.getName())
            .log(java.util.logging.Level.SEVERE, null, ex);
84     } catch (javax.swing.UnsupportedLookAndFeelException ex) {
85         java.util.logging.Logger.getLogger(Simbolo.class.getName())
            .log(java.util.logging.Level.SEVERE, null, ex);
86     }
87     //</editor-fold>
88
89     /* Create and display the form */
90     java.awt.EventQueue.invokeLater(new Runnable() {
91         public void run() {
92             // new Simbolo().setVisible(true);
93         }
94     });
95 }
96 // Variables declaration - do not modify//GEN-BEGIN:variables
97 // End of variables declaration//GEN-END:variables
98 }

```

### 2.6.7. Terminal.java

```

1  //SimAS / Gramatica
2  //Terminal
3
4  package es.uco.simas.util.gramatica;
5
6  /**
7   * @author vanesa
8   */
9  public class Terminal extends Simbolo{
10
11     public Terminal(String nombre, String valor) {
12         super(nombre, valor);
13         initComponents();
14     }
15
16     /**
17      * This method is called from within the constructor to initialize
18      * the form.
19      * WARNING: Do NOT modify this code. The content of this method is
20      * always
21      * regenerated by the Form Editor.
22      */
23     @SuppressWarnings("unchecked")
24     // <editor-fold defaultstate="collapsed" desc="Generated Code"> //
25     GEN-BEGIN: initComponents
26     private void initComponents() {
27
28         setDefaultCloseOperation(javax.swing.WindowConstants.
29             EXIT_ON_CLOSE);

```

```
26 |
27 |     javax.swing.GroupLayout layout = new javax.swing.GroupLayout(
28 |         getContentPane());
29 |     getContentPane().setLayout(layout);
30 |     layout.setHorizontalGroup(
31 |         layout.createParallelGroup(javax.swing.GroupLayout.
32 |             Alignment.LEADING)
33 |         .addGap(0, 400, Short.MAX_VALUE)
34 |     );
35 |     layout.setVerticalGroup(
36 |         layout.createParallelGroup(javax.swing.GroupLayout.
37 |             Alignment.LEADING)
38 |         .addGap(0, 300, Short.MAX_VALUE)
39 |     );
40 |     pack();
41 | }// </editor-fold>//GEN-END: initComponents
42 |
43 | // Variables declaration - do not modify//GEN-BEGIN: variables
44 | // End of variables declaration//GEN-END: variables
45 | }
```



# Referencias

1. Descarga de **Netbeans**:  
Último acceso: junio de 2015.  
URL: <http://netbeans.org/downloads>
2. Descarga de **Java**:  
Último acceso: junio de 2015.  
URL: <https://www.java.com/es/download/>
3. Tutorial de **Java Swing**:  
Último acceso: junio de 2015.  
URL: <http://docs.oracle.com/javase/tutorial/uiswing/>
4. **Itext**:  
Último acceso: junio de 2015.  
URL: <http://itextpdf.com/>
5. **XML**:  
Último acceso: junio de 2015.  
URL: <http://www.w3.org/XML/>

