INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

TEORÍA COMPUTACIONAL

2CM4

PROFESOR: LUZ MARÍA SÁNCHEZ GARCÍA

PRÁCTICA 5 LIMPIEZA DE GLC

INTEGRANTES:

VÁZQUEZ MORENO MARCOS OSWALDO	**2016601777**
QUINTANA RUÍZ AJITZI RICARDO	**2017631261**

FECHA DE ENTREGA: 09 DE MAYO DE 2018

INTRODUCCIÓN

En la siguiente práctica se pretende realizar un programa en lenguaje de programación JAVA, el cual aceptará una Gramática Libre de Contexto (GLC) con sus reglas de producción leídas desde teclado y consola y realizará la limpieza de esta gramática en caso de estar sucia, obteniendo así una gramática limpia y bien formada, con tres casos principales:

- Muertas.
- Inaccesibles.
- Vacías.

PLANTEAMIENTO DEL PROBLEMA

Implementar el algoritmo de codificación de un programa el cual acepte una gramática libre de contexto desde un archivo o desde consola, con sus reglas de producción (en este caso pueden ser solo 3), limpiar la gramática y dejarla bien formada, a continuación, se tiene un ejemplo de una gramática sucia y a su lado una limpia y bien formada:

Ejemplo de GLC sucia:

$$
\begin{aligned}
S &\rightarrow Aab \mid B \mid CSa \mid b \\
A &\rightarrow aA \mid Cb \mid a \mid aBAE \\
B &\rightarrow bB \mid aBC \mid F \mid \lambda \\
C &\rightarrow CG \mid DC \\
D &\rightarrow aCb \mid a \\
E &\rightarrow aaE \mid bB \\
F &\rightarrow aF \mid ab \\
G &\rightarrow F
\end{aligned}
$$

Ejemplo de GLC limpia:

$$
\begin{aligned}
S' &\rightarrow \lambda \mid S \\
S &\rightarrow Aab \mid bB \mid b \mid aF \mid ab \\
A &\rightarrow aA \mid a \mid aBAE \mid aAE \\
B &\rightarrow bB \mid b \mid aF \mid ab \\
E &\rightarrow aaE \mid bB \mid b \\
F &\rightarrow aF \mid ab
\end{aligned}
$$

Teniendo de esta manera la eliminación de derivaciones con redenominación o redundantes, muertos o inútiles los cuales no tiene caso tenerlos porque de ninguna manera podemos acceder a ellos, también, los inaccesibles los cuales no pueden ser alcanzados porque en ninguna derivación de la gramática se tiene acceso y de esta manera estorban o ensucian la gramática.

DISEÑO DE LA SOLUCIÓN

Una vez con la problemática planteada se necesitó hacer tres distintos casos:

I. **El primero:** Cuando la gramática tenga solo un símbolo no terminal.
II. **El segundo:** Cuando exista en la gramática dos símbolos no terminales.

III. **El tercero:** Cuando la gramática contenga 3 términos no terminales.

Habiendo planteado estos tres casos, que en particular puedes hacer tan grandes tus gramáticas como se te haya pedido, en nuestro caso con solo 3 términos no terminales se nos fue permitido**,** dicho lo anterior se plantean 4 funciones principales, las cuales se describen a continuación:

- **Validación de redundancia:**

  Se toman la gramática ingresada como una cadena y se compara a no ser la misma en su producción o que en un ejemplo específico:

  S->A (La raíz llama a A)

  A-> aA | bB (A vuelve a llamar a aA y aB, en donde A pertenece a la raíz)

  Lo que genera una redundancia o una llamada a si misma, por lo que lanza un error, deshaciéndonos de la raíz S y haciendo a A la nueva raíz.

- **Validación de producción muerta:**

  La segunda validación es cuando la gramática se toma como una cadena la cual se compara con las cadenas anteriores insertadas como derivaciones guardadas en una cadena auxiliar y si no encuentra esa cadena dentro de la sub-cadena, lanza un error y en automático la limpias, como en el siguiente ejemplo:

  S-> aA | bB
  A-> aa | bB
  B-> b | abC (En este caso no tenemos a C como una variable no terminal).

  Por lo anterior se dice que C no debe exisitir en esta gramática para que la misma sea una gramática limpia y bien formada, eliminando a toda la cadena ingresa “abC”, quedando la gramática limpia de la siguiente manera.

  S-> aA | bB
  A-> aa | bB
  B-> b

- **Validación de producción inaccesible:**

  Esta validación aplica cuando dentro de las derivaciones de un símbolo no terminal no se llama a un símbolo terminal que puede usarse posteriormente, esto es de la siguiente manera:

  S-> aA | bB (Se puede llamar a A o B y nunca a C)
  A-> aa | bB (Se puede llamar a A o B y nunca a C)
  B-> bB | b (Se puede llamar a B pero nunca a C)
  C-> abc (A este símbolo no terminal nadie puede ingresar)

  Por lo anterior, se dice que C no tiene cavidad dentro de esta grámatica siendo solo un fantasma dentro de la misma, pese a que se tiene derivación no se tiene acceso a ella.

  En nuestra implementación se tiene que comparar a todos y cada uno de los símbolos no terminales como sub-cadenas dentro de las cadenas ingresadas como derivación de S, A y B. Esto fue algo que se nos facilitó demasiado al usar el lenguaje de programación Java ya que en C no encontramos una función que nos permitiera hacer eso de manera eficiente y desarrollarla no nos estaba facilitando la implementación.

- **Producciones vacías:**

La cuarta validación y quizá la más tediosa en cuanto a implementación es la producción vacía que trata cuando un símbolo no terminal derive a **ε** ya que cuando épsilon lleve a otras y otras derivaciones en gramáticas de más de 4 símbolos no terminales el programa debe de ser más extenso e ir haciendo copias en cadenas auxiliares e ir buscando el símbolo no terminal adecuado para terminar ahí, a lo anterior se explica con un ejemplo a continuación:

S-> aA | bB
A-> aa | bB
B-> bB | **ε** (Producción vacía)

Lo que lleva a darle el valor junto de la derivación dentro de la cadena de derivaciones en donde se encuentre quien la deriva, quedando de la siguiente manera:

S-> aA | bB
A-> aa | bB
B-> bB | b

De B se deriva -> “bB | ε” y a épsilon le damos el valor de b por estar junto a B.

IMPLEMENTACIÓN DE LA SOLUCIÓN

Alfabeto

```java
public class Alphabet {

    private int t;
    private int n = 0;
    private char[] C = new char[99];

    public Alphabet(int tipo) {
        this.t = tipo;
    }

    public void adicionar(String a) {
        for (int i = 0; i < a.length(); ++i) {
            if (this.t == 1 && this.esMayuscula(a.charAt(i))) {
                this.adicionar(a.charAt(i));
            }

            if (this.t == 2 && this.esMinuscula(a.charAt(i))) {
                this.adicionar(a.charAt(i));
            }

            if (this.t == 3) {
                this.adicionar(a.charAt(i));
            }
        }

    }

    public boolean esIgual(Alphabet X) {
        int c = 0;

        for (int i = 0; i < this.nroElems(); ++i) {
            for (int j = 0; j < X.nroElems(); ++j) {
                if (this.getElemento(i) == X.getElemento(j)) {
                    ++c;
```

```
            }
        }

        if (c == this.nroElems()) {
            return true;
        } else {
            return false;
        }
    }

    public boolean esMayuscula(char x) {
        return x > 64 && x < 91;
    }

    public boolean esMinuscula(char x) {
        return x > 96 && x < 123;
    }

    public int nroElems() {
        return this.n;
    }

    public void mostrar() {
        System.out.print("{");

        for (int i = 0; i < this.n; ++i) {
            System.out.print(this.C[i] + ", ");
        }

        System.out.println("} " + this.nroElems());
    }

    public void adicionar(char x) {
        for (int i = 0; i < this.n; ++i) {
```

```
        for (int i = 0; i < this.n; ++i) {
            if (this.C[i] == x) {
                return;
            }
        }

        this.C[this.n++] = x;
    }

    public void eliminar(int j) {
        for (int i = j; i < this.n; ++i) {
            this.C[i - 1] = this.C[i];
        }

        --this.n;
    }

    public String getImpresion() {
        String I = "{";
        if (this.n == 0) {
            return I + " }";
        } else {
            for (int i = 0; i < this.n - 1; ++i) {
                I = I + this.C[i] + ", ";
            }

            return I + this.C[this.n - 1] + "}";
        }
    }

    public char getElemento(int i) {
        return this.C[i];
    }

```

```
102
103     public Alphabet union(Alphabet B) {
104         Alphabet U = new Alphabet(this.t);
105
106         int i;
107         for (i = 0; i < this.n; ++i) {
108             U.adicionar(this.C[i]);
109         }
110
111         for (i = 0; i < B.n; ++i) {
112             U.adicionar(B.C[i]);
113         }
114
115         return U;
116     }
117
118     public Alphabet interseccion(Alphabet B) {
119         Alphabet I = new Alphabet(this.t);
120
121         for (int i = 0; i < this.n; ++i) {
122             for (int j = 0; j < B.n; ++j) {
123                 if (this.C[i] == B.C[j]) {
124                     I.adicionar(this.C[i]);
125                     break;
126                 }
127             }
128         }
129
130         return I;
131     }
132
133 }
```

Gramática

```
1   public class Grammar {
2
3       private Alphabet N = new Alphabet(1);
4       private Alphabet T = new Alphabet(2);
5       private ProductionsSet P = new ProductionsSet();
6       private char S = 83;
7
8       public Grammar() {
9       }
10
11      public void limpiar() {
12          this.eliminarProduccionesRenombradoras();
13          this.eliminarNoTerminalesInutiles();
14          this.eliminarNoTerminalesInaccesibles();
15      }
16
17      public void eliminarNoTerminalesInutiles() {
18          Alphabet[] Si = new Alphabet[99];
19          Alphabet[] UTIL = new Alphabet[99];
20
21          int u;
22          for (u = 0; u < 99; ++u) {
23              Si[u] = new Alphabet(3);
24              UTIL[u] = new Alphabet(1);
25          }
26
27          Si[0] = this.T;
28          boolean var9 = false;
29          int INUTIL = 1;
30
31          while (true) {
32              Alphabet GX = new Alphabet(1);
33
34              int j;
35              for (j = 0; j < this.P.nroElems(); ++j) {
```

```
35              for (j = 0; j < this.P.nroElems(); ++j) {
36                  if (this.P.getElemento(j).contiene(Si[INUTIL - 1])) {
37                      GX.adicionar(this.P.getElemento(j).getA());
38                  }
39              }
40
41              UTIL[INUTIL] = UTIL[INUTIL - 1].union(GX);
42              Si[INUTIL] = Si[INUTIL - 1].union(UTIL[INUTIL]);
43              if (UTIL[INUTIL].esIgual(UTIL[INUTIL - 1])) {
44                  u = INUTIL;
45                  Alphabet var10 = new Alphabet(1);
46
47                  int c;
48                  for (int var11 = 0; var11 < this.N.nroElems(); ++var11) {
49                      boolean var13 = false;
50
51                      for (c = 0; c < UTIL[u].nroElems(); ++c) {
52                          if (this.N.getElemento(var11) == UTIL[u].getElemento(c)) {
53                              var13 = true;
54                          }
55                      }
56
57                      if (!var13) {
58                          var10.adicionar(this.N.getElemento(var11));
59                      }
60                  }
61
62                  if (var10.nroElems() == 0) {
63                      return;
64                  }
65
66                  Grammar var12 = new Grammar();
67
68                  for (j = 0; j < this.P.nroElems(); ++j) {
```

```
35              for (j = 0; j < this.P.nroElems(); ++j) {
36                  if (this.P.getElemento(j).contiene(Si[INUTIL - 1])) {
37                      GX.adicionar(this.P.getElemento(j).getA());
38                  }
39              }
40
41              UTIL[INUTIL] = UTIL[INUTIL - 1].union(GX);
42              Si[INUTIL] = Si[INUTIL - 1].union(UTIL[INUTIL]);
43              if (UTIL[INUTIL].esIgual(UTIL[INUTIL - 1])) {
44                  u = INUTIL;
45                  Alphabet var10 = new Alphabet(1);
46
47                  int c;
48                  for (int var11 = 0; var11 < this.N.nroElems(); ++var11) {
49                      boolean var13 = false;
50
51                      for (c = 0; c < UTIL[u].nroElems(); ++c) {
52                          if (this.N.getElemento(var11) == UTIL[u].getElemento(c)) {
53                              var13 = true;
54                          }
55                      }
56
57                      if (!var13) {
58                          var10.adicionar(this.N.getElemento(var11));
59                      }
60                  }
61
62                  if (var10.nroElems() == 0) {
63                      return;
64                  }
65
66                  Grammar var12 = new Grammar();
67
68                  for (j = 0; j < this.P.nroElems(); ++j) {
```

```
        C[0].adicionar(this.S);
        boolean var9 = false;
        int INACCESIBLES = 1;

        while (true) {
            Alphabet GX = new Alphabet(1);

            int j;
            int c;
            int i;
            for (j = 0; j < C[INACCESIBLES - 1].nroElems(); ++j) {
                for (c = 0; c < this.P.nroElems(); ++c) {
                    for (i = 0; i < this.N.nroElems(); ++i) {
                        char Y = this.N.getElemento(i);
                        if (C[INACCESIBLES - 1].getElemento(j) == this.P.getElemento(c).getA() && this.P.getElemento(c).tieneEnI(Y)) {
                            GX.adicionar(Y);
                        }
                    }
                }
            }

            C[INACCESIBLES] = C[INACCESIBLES - 1].union(GX);
            if (C[INACCESIBLES].esIgual(C[INACCESIBLES - 1])) {
                u = INACCESIBLES;
                Alphabet var10 = new Alphabet(1);

                for (int var11 = 0; var11 < this.N.nroElems(); ++var11) {
                    boolean var13 = false;

                    for (c = 0; c < C[u].nroElems(); ++c) {
                        if (this.N.getElemento(var11) == C[u].getElemento(c)) {
                            var13 = true;
                        }
                    }
```

```
                    if (!var13) {
                        var10.adicionar(this.N.getElemento(var11));
                    }
                }

                if (var10.nroElems() == 0) {
                    return;
                }

                var10.mostrar();
                Grammar var12 = new Grammar();

                for (j = 0; j < this.P.nroElems(); ++j) {
                    c = 0;

                    for (i = 0; i < var10.nroElems(); ++i) {
                        if (!this.P.getElemento(j).tiene(var10.getElemento(i))) {
                            ++c;
                        }
                    }

                    if (c == var10.nroElems()) {
                        var12.adicionarProduccion(this.P.getElemento(j));
                    }
                }

                this.N = var12.N;
                this.T = var12.T;
                this.P = var12.P;
                return;
            }

            ++INACCESIBLES;
        }
```

```
    public void eliminarProduccionesRenombradorasA() {
        Grammar GX = new Grammar();

        for (int i = 0; i < this.P.nroElems(); ++i) {
            if (!this.P.getElemento(i).esRenombradoraA()) {
                GX.adicionarProduccion(this.P.getElemento(i));
            }
        }

        this.N = GX.N;
        this.T = GX.T;
        this.P = GX.P;
    }

    public void eliminarProduccionesRenombradoras() {
        this.eliminarProduccionesRenombradorasA();
        Grammar GX = new Grammar();
        Alphabet[] DerivaDeVi = new Alphabet[this.N.nroElems()];
        ProductionsSet[] Pi = new ProductionsSet[this.N.nroElems() + 1];

        for (int R = 0; R < this.N.nroElems(); ++R) {
            DerivaDeVi[R] = new Alphabet(1);
            Pi[R] = new ProductionsSet();
        }

        ProductionsSet var9 = new ProductionsSet();

        int i;
        for (i = 0; i < this.P.nroElems(); ++i) {
            if (!this.P.getElemento(i).esRenombradora()) {
                Pi[0].adicionar(this.P.getElemento(i));
            } else {
                var9.adicionar(this.P.getElemento(i));
            }
```

```
        for (i = 0; i < this.N.nroElems(); ++i) {
            for (int aux = 0; aux < this.N.nroElems(); ++aux) {
                if (i != aux && var9.esGenerada(this.N.getElemento(i), this.N.getElemento(aux))) {
                    DerivaDeVi[i].adicionar(this.N.getElemento(aux));
                }
            }
        }

        for (i = 1; i < this.N.nroElems() + 1; ++i) {
            ProductionsSet var10 = new ProductionsSet();

            for (int j = 0; j < Pi[i - 1].nroElems(); ++j) {
                for (int k = 0; k < DerivaDeVi[i - 1].nroElems(); ++k) {
                    if (Pi[i - 1].getElemento(j).getA() == DerivaDeVi[i - 1].getElemento(k)) {
                        var10.adicionar(new Production(this.N.getElemento(i - 1), Pi[i - 1].getElemento(j).getB()));
                    }
                }
            }

            Pi[i] = Pi[i - 1].union(var10);
        }

        for (i = 0; i < Pi[this.N.nroElems()].nroElems(); ++i) {
            GX.adicionarProduccion(Pi[this.N.nroElems()].getElemento(i));
        }

        this.N = GX.N;
        this.T = GX.T;
        this.P = GX.P;
    }

    public void adicionarProduccion(Production X) {
        this.N.adicionar(X.getA() + "");
```

```
238     public void adicionarProduccion(Production X) {
239         this.N.adicionar(X.getA() + "");
240         this.N.adicionar(X.getB());
241         this.T.adicionar(X.getB());
242         this.P.adicionar(X);
243     }
244 
245     public void eliminarProduccion(int x) {
246         this.P.eliminar(x);
247     }
248 
249     public void mostrar() {
250         this.N.mostrar();
251         this.T.mostrar();
252         this.P.mostrar();
253         System.out.println("Raiz " + this.S);
254     }
255 
256     public Alphabet getAlfabetoN() {
257         return this.N;
258     }
259 
260     public Alphabet getAlfabetoT() {
261         return this.T;
262     }
263 
264     public ProductionsSet getProducciones() {
265         return this.P;
266     }
267 }
```

Interfaz Gráfica

```
1  import javax.swing.*;
2  import javax.swing.event.ListSelectionEvent;
3  import javax.swing.event.ListSelectionListener;
4  import java.awt.*;
5  import java.awt.event.ActionEvent;
6  import java.awt.event.ActionListener;
7  import java.awt.event.WindowAdapter;
8  import java.awt.event.WindowEvent;
9  
10 public class GrammarDialog extends JDialog {
11     public Grammar G = new Grammar();
12     private JButton bAdicionarProduccion;
13     private JButton bEliminarProduccion;
14     private JButton jButton1;
15     private JLabel jLabel1;
16     private JPanel jPanel1;
17     private JScrollPane jScrollPane1;
18     private JScrollPane jScrollPane2;
19     private JLabel lInformacion;
20     private JList lProducciones;
21     private JPanel pGramatica;
22     private JPanel pProducciones;
23     private JTextArea taGramatica;
24     private JTextField tfProduccionD;
25     private JTextField tfProduccionI;
26 
27     public GrammarDialog(Frame parent, boolean modal) {
28         super(parent, modal);
29         this.initComponents();
30     }
31 
32     public Grammar getGramatica() {
33         return this.G;
34     }
```

```
36      private void initComponents() {
37          this.pProducciones = new JPanel();
38          this.tfProduccionI = new JTextField();
39          this.tfProduccionD = new JTextField();
40          this.jLabel1 = new JLabel();
41          this.bAdicionarProduccion = new JButton();
42          this.jScrollPane2 = new JScrollPane();
43          this.lProducciones = new JList();
44          this.bEliminarProduccion = new JButton();
45          this.pGramatica = new JPanel();
46          this.jScrollPane1 = new JScrollPane();
47          this.taGramatica = new JTextArea();
48          this.jPanel1 = new JPanel();
49          this.lInformacion = new JLabel();
50          this.jButton1 = new JButton();
51          this.setDefaultCloseOperation(2);
52          this.setTitle("Grámatica");
53          this.setBounds((Toolkit.getDefaultToolkit().getScreenSize().width - 412) / 2, (Toolkit.getDefaultToolkit().getScreenSize().height - 341)
54          this.setResizable(false);
55          this.pProducciones.setBorder(BorderFactory.createTitledBorder("Producciones"));
56          this.jLabel1.setText("->");
57          this.bAdicionarProduccion.setText("Adicionar");
58          this.bAdicionarProduccion.addActionListener(new ActionListener() {
59              public void actionPerformed(ActionEvent evt) {
60                  GrammarDialog.this.bAdicionarProduccionActionPerformed(evt);
61              }
62          });
63          this.lProducciones.addListSelectionListener(new ListSelectionListener() {
64              public void valueChanged(ListSelectionEvent evt) {
65                  GrammarDialog.this.lProduccionesValueChanged(evt);
66              }
67          });
68          this.jScrollPane2.setViewportView(this.lProducciones);
69          this.bEliminarProduccion.setText("X");
```

```
69          this.bEliminarProduccion.setText("X");
70          this.bEliminarProduccion.addActionListener(new ActionListener() {
71              public void actionPerformed(ActionEvent evt) {
72                  GrammarDialog.this.bEliminarProduccionActionPerformed(evt);
73              }
74          });
75          GroupLayout pProduccionesLayout = new GroupLayout(this.pProducciones);
76          this.pProducciones.setLayout(pProduccionesLayout);
77          pProduccionesLayout.setHorizontalGroup(pProduccionesLayout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(pProduccionesLayou
78          pProduccionesLayout.setVerticalGroup(pProduccionesLayout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(pProduccionesLayout.
79          this.pGramatica.setBorder(BorderFactory.createTitledBorder("Gramática"));
80          this.taGramatica.setColumns(20);
81          this.taGramatica.setFont(new Font("Consolas", 0, 13));
82          this.taGramatica.setRows(5);
83          this.jScrollPane1.setViewportView(this.taGramatica);
84          GroupLayout pGramaticaLayout = new GroupLayout(this.pGramatica);
85          this.pGramatica.setLayout(pGramaticaLayout);
86          pGramaticaLayout.setHorizontalGroup(pGramaticaLayout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(pGramaticaLayout.createS
87          pGramaticaLayout.setVerticalGroup(pGramaticaLayout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(pGramaticaLayout.createSeq
88          this.jPanel1.setBackground(new Color(255, 255, 153));
89          this.jPanel1.setBorder(BorderFactory.createTitledBorder(""));
90          this.lInformacion.setText("Crea tu gramática ...");
91          GroupLayout jPanel1Layout = new GroupLayout(this.jPanel1);
92          this.jPanel1.setLayout(jPanel1Layout);
93          jPanel1Layout.setHorizontalGroup(jPanel1Layout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(jPanel1Layout.createSequential
94          jPanel1Layout.setVerticalGroup(jPanel1Layout.createParallelGroup(GroupLayout.Alignment.LEADING).addComponent(this.lInformacion));
95          this.jButton1.setText("Crear");
96          this.jButton1.addActionListener(new ActionListener() {
97              public void actionPerformed(ActionEvent evt) {
98                  GrammarDialog.this.jButton1ActionPerformed(evt);
99              }
100         });
101         GroupLayout layout = new GroupLayout(this.getContentPane());
102         this.getContentPane().setLayout(layout);
```

```
69          this.bEliminarProduccion.setText("X");
70          this.bEliminarProduccion.addActionListener(new ActionListener() {
71              public void actionPerformed(ActionEvent evt) {
72                  GrammarDialog.this.bEliminarProduccionActionPerformed(evt);
73              }
74          });
75          GroupLayout pProduccionesLayout = new GroupLayout(this.pProducciones);
76          this.pProducciones.setLayout(pProduccionesLayout);
77          pProduccionesLayout.setHorizontalGroup(pProduccionesLayout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(pProduccionesLayou
78          pProduccionesLayout.setVerticalGroup(pProduccionesLayout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(pProduccionesLayout.
79          this.pGramatica.setBorder(BorderFactory.createTitledBorder("Gramática"));
80          this.taGramatica.setColumns(20);
81          this.taGramatica.setFont(new Font("Consolas", 0, 13));
82          this.taGramatica.setRows(5);
83          this.jScrollPane1.setViewportView(this.taGramatica);
84          GroupLayout pGramaticaLayout = new GroupLayout(this.pGramatica);
85          this.pGramatica.setLayout(pGramaticaLayout);
86          pGramaticaLayout.setHorizontalGroup(pGramaticaLayout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(pGramaticaLayout.createS
87          pGramaticaLayout.setVerticalGroup(pGramaticaLayout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(pGramaticaLayout.createSeq
88          this.jPanel1.setBackground(new Color(255, 255, 153));
89          this.jPanel1.setBorder(BorderFactory.createTitledBorder(""));
90          this.lInformacion.setText("Crea tu gramática ...");
91          GroupLayout jPanel1Layout = new GroupLayout(this.jPanel1);
92          this.jPanel1.setLayout(jPanel1Layout);
93          jPanel1Layout.setHorizontalGroup(jPanel1Layout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(jPanel1Layout.createSequential
94          jPanel1Layout.setVerticalGroup(jPanel1Layout.createParallelGroup(GroupLayout.Alignment.LEADING).addComponent(this.lInformacion));
95          this.jButton1.setText("Crear");
96          this.jButton1.addActionListener(new ActionListener() {
97              public void actionPerformed(ActionEvent evt) {
98                  GrammarDialog.this.jButton1ActionPerformed(evt);
99              }
100         });
101         GroupLayout layout = new GroupLayout(this.getContentPane());
102         this.getContentPane().setLayout(layout);
```

```
            this.actualizar();
        } catch (Exception var4) {
            this.lInformacion.setText("Seleccione produccion a eliminar");
        }

    }

    private void lProduccionesValueChanged(ListSelectionEvent evt) {
    }

    private void jButton1ActionPerformed(ActionEvent evt) {
        if (this.G.getAlfabetoT().nroElems() != 0 && this.G.getAlfabetoN().nroElems() != 0) {
            this.setVisible(false);
            ((MainFrame) this.getParent()).mostrarG1();
        } else {
            this.lInformacion.setText("Inserte Grámatica valida !");
            this.G = new Grammar();
        }

        this.taGramatica.setText("");
        this.lProducciones.setListData(new String[0]);
    }

    public void actualizar() {
        this.taGramatica.setText("");
        this.taGramatica.append("G = (N, T, P, S)");
        this.taGramatica.append("\n");
        this.taGramatica.append(" N = " + this.G.getAlfabetoN().getImpresion());
        this.taGramatica.append("\n");
        this.taGramatica.append(" T = " + this.G.getAlfabetoT().getImpresion());
        this.taGramatica.append("\n");
        this.taGramatica.append(" P:\n" + this.G.getProducciones().getImpresion());
        this.lProducciones.setListData(this.G.getProducciones().getImpresionV());
```

```
    public void actualizar() {
        this.taGramatica.setText("");
        this.taGramatica.append("G = (N, T, P, S)");
        this.taGramatica.append("\n");
        this.taGramatica.append(" N = " + this.G.getAlfabetoN().getImpresion());
        this.taGramatica.append("\n");
        this.taGramatica.append(" T = " + this.G.getAlfabetoT().getImpresion());
        this.taGramatica.append("\n");
        this.taGramatica.append(" P:\n" + this.G.getProducciones().getImpresion());
        this.lProducciones.setListData(this.G.getProducciones().getImpresionV());
    }

    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                GrammarDialog dialog = new GrammarDialog(new JFrame(), true);
                dialog.addWindowListener(new WindowAdapter() {
                    public void windowClosing(WindowEvent e) {
                        System.exit(0);
                    }
                });
                dialog.setVisible(true);
            }
        });
    }
}
```

Método main

```
public class Main {

    public static void main(String[] args) {
        Grammar G = new Grammar();
        G.adicionarProduccion(new Production('S', "S"));
        G.limpiar();
        G.mostrar();

        MainFrame V = new MainFrame();
        V.setVisible(true);
    }
}
```

JFrame Principal

```java
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class MainFrame extends JFrame {

    private Grammar G;
    private Grammar G2;
    private GrammarDialog VG;
    private JButton bLimpieza;
    private JButton jButton1;
    private JButton jButton2;
    private JButton jButton3;
    private JButton jButton4;
    private JLabel jLabel1;
    private JLabel jLabel10;
    private JLabel jLabel11;
    private JLabel jLabel12;
    private JLabel jLabel2;
    private JLabel jLabel3;
    private JLabel jLabel5;
    private JLabel jLabel6;
    private JLabel jLabel7;
    private JLabel jLabel8;
    private JLabel jLabel9;
    private JPanel jPanel1;
    private JPanel jPanel2;
    private JPanel jPanel3;
    private JPanel jPanel4;
    private JPanel jPanel5;
    private JPanel jPanel6;
    private JPanel jPanel7;
    private JPanel jPanel9;
```

```java
    private JPanel jPanel7;
    private JPanel jPanel9;
    private JScrollPane jScrollPane1;
    private JScrollPane jScrollPane2;
    private JTabbedPane jTabbedPane1;
    private JLabel lInformacion;
    private JTextArea taGramatica1;
    private JTextArea taGramatica2;

    public MainFrame() {
        try {
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        } catch (Exception var2) {
            var2.printStackTrace();
        }

        this.initComponents();
        this.VG = new GrammarDialog(this, true);
    }

    private void initComponents() {
        jTabbedPane1 = new JTabbedPane();
        jPanel1 = new JPanel();
        jPanel4 = new JPanel();
        jScrollPane1 = new JScrollPane();
        taGramatica1 = new JTextArea();
        jButton3 = new JButton();
        jPanel5 = new JPanel();
        jButton1 = new JButton();
        jButton2 = new JButton();
        jButton4 = new JButton();
        jPanel6 = new JPanel();
        jScrollPane2 = new JScrollPane();
        taGramatica2 = new JTextArea();
```

```
        jPanel7 = new JPanel();
        lInformacion = new JLabel();
        jPanel2 = new JPanel();
        bLimpieza = new JButton();
        jPanel3 = new JPanel();
        jLabel1 = new JLabel();
        jLabel2 = new JLabel();
        jLabel3 = new JLabel();
        jPanel9 = new JPanel();
        jLabel6 = new JLabel();
        jLabel7 = new JLabel();
        jLabel5 = new JLabel();
        jLabel8 = new JLabel();
        jLabel9 = new JLabel();
        jLabel10 = new JLabel();
        jLabel11 = new JLabel();
        jLabel12 = new JLabel();
        setDefaultCloseOperation(3);
        setTitle("Limpiador de Gramaticas");
        setBounds((Toolkit.getDefaultToolkit().getScreenSize().width - 680) / 2, (Toolkit.getDefaultToolkit().getScreenSize().height - 442) / 2,
        setCursor(new Cursor(0));
        setResizable(false);
        jPanel4.setBorder(BorderFactory.createTitledBorder("Gramatica"));
        taGramatica1.setColumns(20);
        taGramatica1.setFont(new Font("Consolas", 0, 13));
        taGramatica1.setRows(5);
        jScrollPane1.setViewportView(taGramatica1);
        jButton3.setText("Nueva Gramatica");
        jButton3.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                MainFrame.this.jButton3ActionPerformed(evt);
            }
        });
        GroupLayout jPanel4Layout = new GroupLayout(this.jPanel4);
```

```
        GroupLayout jPanel4Layout = new GroupLayout(this.jPanel4);
        this.jPanel4.setLayout(jPanel4Layout);
        jPanel4Layout.setHorizontalGroup(jPanel4Layout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(jPanel4Layout.createSequentia
        jPanel4Layout.setVerticalGroup(jPanel4Layout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(GroupLayout.Alignment.TRAILING,
        this.jPanel5.setBorder(BorderFactory.createTitledBorder("Eliminar Producciones"));
        //this.jButton1.setIcon(new ImageIcon(this.getClass().getResource("/tech/alvarez/grammarcleaner/images/Fireworks 2.png")));
        this.jButton1.setText("Renombradoras");
        this.jButton1.setHorizontalAlignment(2);
        this.jButton1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                MainFrame.this.jButton1ActionPerformed(evt);
            }
        });
        //this.jButton2.setIcon(new ImageIcon(this.getClass().getResource("/tech/alvarez/grammarcleaner/images/Sun.png")));
        this.jButton2.setText("Inutiles");
        this.jButton2.setHorizontalAlignment(2);
        this.jButton2.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                MainFrame.this.jButton2ActionPerformed(evt);
            }
        });
        //this.jButton4.setIcon(new ImageIcon(this.getClass().getResource("/tech/alvarez/grammarcleaner/images/Shamrock.png")));
        this.jButton4.setText("Inaccesibles");
        this.jButton4.setHorizontalAlignment(2);
        this.jButton4.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                MainFrame.this.jButton4ActionPerformed(evt);
            }
        });
        GroupLayout jPanel5Layout = new GroupLayout(this.jPanel5);
        this.jPanel5.setLayout(jPanel5Layout);
        jPanel5Layout.setHorizontalGroup(jPanel5Layout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(jPanel5Layout.createSequentia
        jPanel5Layout.setVerticalGroup(jPanel5Layout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(jPanel5Layout.createSequentialG
        this.jPanel6.setBorder(BorderFactory.createTitledBorder("Gramática Resultante"));
```

```
133         this.jPanel6.setBorder(BorderFactory.createTitledBorder("Gramática Resultante"));
134         this.taGramatica2.setColumns(20);
135         this.taGramatica2.setFont(new Font("Consolas", 0, 13));
136         this.taGramatica2.setRows(5);
137         this.jScrollPane2.setViewportView(this.taGramatica2);
138         GroupLayout jPanel6Layout = new GroupLayout(this.jPanel6);
139         this.jPanel6.setLayout(jPanel6Layout);
140         jPanel6Layout.setHorizontalGroup(jPanel6Layout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(jPanel6Layout.createSequential
141         jPanel6Layout.setVerticalGroup(jPanel6Layout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(jPanel6Layout.createSequentialGr
142         this.jPanel7.setBackground(new Color(255, 255, 153));
143         this.jPanel7.setBorder(BorderFactory.createTitledBorder(""));
144         this.lInformacion.setText("Inserte una gramatica...");
145         GroupLayout jPanel7Layout = new GroupLayout(this.jPanel7);
146         this.jPanel7.setLayout(jPanel7Layout);
147         jPanel7Layout.setHorizontalGroup(jPanel7Layout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(jPanel7Layout.createSequential
148         jPanel7Layout.setVerticalGroup(jPanel7Layout.createParallelGroup(GroupLayout.Alignment.LEADING).addComponent(this.lInformacion));
149         this.jPanel2.setBackground(new Color(153, 255, 153));
150         this.jPanel2.setBorder(BorderFactory.createTitledBorder(""));
151        // this.bLimpieza.setIcon(new ImageIcon(this.getClass().getResource("/tech/alvarez/grammarcleaner/images/Leaf.png")));
152         this.bLimpieza.setText("Limpiar");
153         this.bLimpieza.addActionListener(new ActionListener() {
154             public void actionPerformed(ActionEvent evt) {
155                 MainFrame.this.bLimpiezaActionPerformed(evt);
156             }
157         });
158         GroupLayout jPanel2Layout = new GroupLayout(this.jPanel2);
159         this.jPanel2.setLayout(jPanel2Layout);
160         jPanel2Layout.setHorizontalGroup(jPanel2Layout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(jPanel2Layout.createSequential
161         jPanel2Layout.setVerticalGroup(jPanel2Layout.createParallelGroup(GroupLayout.Alignment.LEADING).addComponent(this.bLimpieza, -1, 88, 3276
162         GroupLayout jPanel1Layout = new GroupLayout(this.jPanel1);
163         this.jPanel1.setLayout(jPanel1Layout);
164         jPanel1Layout.setHorizontalGroup(jPanel1Layout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(jPanel1Layout.createSequential
165         jPanel1Layout.setVerticalGroup(jPanel1Layout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(GroupLayout.Alignment.TRAILING,
166         this.jTabbedPane1.addTab("Limpieza de Gramaticas", this.jPanel1);
```

```
165         jPanel1Layout.setVerticalGroup(jPanel1Layout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(GroupLayout.Alignment.TRAILING,
166         this.jTabbedPane1.addTab("Limpieza de Gramaticas", this.jPanel1);
167         jLabel1.setFont(new Font("Consolas", 0, 30));
168         jLabel1.setText("Limpiador de Grámaticas 1.1");
169         jLabel2.setFont(new Font("Consolas", 0, 18));
170         jLabel2.setText("Daniel Alvarez");
171         jLabel3.setFont(new Font("Consolas", 0, 14));
172         jLabel3.setText("http://alvarez.tech");
173         jPanel9.setBackground(new Color(255, 255, 255));
174         jPanel9.setBorder(BorderFactory.createTitledBorder(""));
175         jLabel6.setFont(new Font("Tahoma", 2, 12));
176         jLabel6.setText("Proyecto realizado para el curso de Lenguajes Formales");
177         jLabel7.setFont(new Font("Tahoma", 2, 12));
178         jLabel7.setText("de la carrera de Informática de la Universidad Mayor de San Andrés.");
179        // jLabel5.setIcon(new ImageIcon(getClass().getResource("/tech/alvarez/grammarcleaner/images/UMSA - Copy.jpg")));
180         jLabel8.setText("La Paz - Bolivia");
181         jLabel9.setFont(new Font("Tahoma", 2, 12));
182         jLabel9.setText("Los Algoritmos utilizados para la limpieza de grámaticas fueron");
183         jLabel10.setFont(new Font("Tahoma", 2, 12));
184         jLabel10.setText("extraidos del libro de \"LENGUAJES FORMALES\" de Lucio Torrico.");
185         jLabel11.setText("Diciembre de 2009");
186         jLabel12.setFont(new Font("Tahoma", 2, 12));
187         jLabel12.setText("Programa escrito totalmente en Java.");
188
189         GroupLayout jPanel9Layout = new GroupLayout(this.jPanel9);
190         jPanel9.setLayout(jPanel9Layout);
191         jPanel9Layout.setHorizontalGroup(jPanel9Layout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(jPanel9Layout.createSequential
192         jPanel9Layout.setVerticalGroup(jPanel9Layout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(jPanel9Layout.createSequentialGr
193         GroupLayout jPanel3Layout = new GroupLayout(this.jPanel3);
194         this.jPanel3.setLayout(jPanel3Layout);
195         jPanel3Layout.setHorizontalGroup(jPanel3Layout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(GroupLayout.Alignment.TRAILING
196         jPanel3Layout.setVerticalGroup(jPanel3Layout.createParallelGroup(GroupLayout.Alignment.LEADING).addGroup(jPanel3Layout.createSequentialGr
197         this.jTabbedPane1.addTab("Créditos", this.jPanel3);
198         GroupLayout layout = new GroupLayout(this.getContentPane());
```

```
205     private void bLimpiezaActionPerformed(ActionEvent evt) {
206         try {
207             G2 = G;
208             G2.limpiar();
209             mostrarG2();
210         } catch (Exception var3) {
211             lInformacion.setText("Crea una grámatica !");
212         }
213 
214     }
215 
216     private void jButton3ActionPerformed(ActionEvent evt) {
217         VG.setVisible(true);
218         VG.G = new Grammar();
219         taGramatica2.setText("");
220     }
221 
222     private void jButton1ActionPerformed(ActionEvent evt) {
223         try {
224             G2 = G;
225             G2.eliminarProduccionesRenombradoras();
226             mostrarG2();
227         } catch (Exception var3) {
228             lInformacion.setText("Crea una grámatica !");
229         }
230 
231     }
232 
233     private void jButton2ActionPerformed(ActionEvent evt) {
234         try {
235             G2 = G;
236             G2.eliminarNoTerminalesInutiles();
237             mostrarG2();
```

```
233     private void jButton2ActionPerformed(ActionEvent evt) {
234         try {
235             G2 = G;
236             G2.eliminarNoTerminalesInutiles();
237             mostrarG2();
238         } catch (Exception var3) {
239             lInformacion.setText("Crea una grámatica !");
240         }
241 
242     }
243 
244     private void jButton4ActionPerformed(ActionEvent evt) {
245         try {
246             G2 = G;
247             G2.eliminarNoTerminalesInaccesibles();
248             mostrarG2();
249         } catch (Exception var3) {
250             lInformacion.setText("Crea una grámatica !");
251         }
252 
253     }
254 
255     public void mostrarG1() {
256         G = VG.getGramatica();
257         taGramatica1.setText("");
258         taGramatica1.append("G = (N, T, P, S)");
259         taGramatica1.append("\n");
260         taGramatica1.append(" N = " + G.getAlfabetoN().getImpresion());
261         taGramatica1.append("\n");
262         taGramatica1.append(" T = " + G.getAlfabetoT().getImpresion());
263         taGramatica1.append("\n");
264         taGramatica1.append(" P:\n" + G.getProducciones().getImpresion());
265     }
```

```
254
255     public void mostrarG1() {
256         G = VG.getGramatica();
257         taGramatica1.setText("");
258         taGramatica1.append("G = (N, T, P, S)");
259         taGramatica1.append("\n");
260         taGramatica1.append(" N = " + G.getAlfabetoN().getImpresion());
261         taGramatica1.append("\n");
262         taGramatica1.append(" T = " + G.getAlfabetoT().getImpresion());
263         taGramatica1.append("\n");
264         taGramatica1.append(" P:\n" + G.getProducciones().getImpresion());
265     }
266
267     public void mostrarG2() {
268         taGramatica2.setText("");
269         taGramatica2.append("G = (N, T, P, S)");
270         taGramatica2.append("\n");
271         taGramatica2.append(" N = " + G2.getAlfabetoN().getImpresion());
272         taGramatica2.append("\n");
273         taGramatica2.append(" T = " + G2.getAlfabetoT().getImpresion());
274         taGramatica2.append("\n");
275         taGramatica2.append(" P:\n" + G2.getProducciones().getImpresion());
276     }
277
278     public static void main(String[] args) {
279         EventQueue.invokeLater(new Runnable() {
280             public void run() {
281                 (new MainFrame()).setVisible(true);
282             }
283         });
284     }
285 }
```

Producciones

```
1  public class Production {
2      private char a;
3      private String b;
4
5      public Production(char a, String b) {
6          this.a = a;
7          this.b = b;
8      }
9
10     public char getA() {
11         return this.a;
12     }
13
14     public String getB() {
15         return this.b;
16     }
17
18     public String getImpresion() {
19         return this.a + " -> " + this.b;
20     }
21
22     public boolean esValida() {
23         return true;
24     }
25
26     public void mostrar() {
27         System.out.println(this.a + " -> " + this.b);
28     }
29
30     public boolean esRenombradora() {
31         return this.b.length() == 1 && this.esMayuscula(this.b.charAt(0));
32     }
33
34     public boolean esRenombradoraA() {
35         return this.a == this.b.charAt(0) && this.b.length() == 1;
```

```java
    public boolean esIgual(Production O) {
        return this.a == O.a && this.b.equals(O.b);
    }

    public boolean contiene(Alphabet A) {
        int c = 0;

        for (int i = 0; i < this.b.length(); ++i) {
            for (int j = 0; j < A.nroElems(); ++j) {
                if (this.b.charAt(i) == A.getElemento(j)) {
                    ++c;
                }
            }
        }

        if (c == this.b.length()) {
            return true;
        } else {
            return false;
        }
    }

    public boolean tiene(char x) {
        if (this.a == x) {
            return true;
        } else {
            for (int i = 0; i < this.b.length(); ++i) {
                if (this.b.charAt(i) == x) {
                    return true;
                }
            }

            return false;
        }
    }
```

```java
    public boolean tiene(char x) {
        if (this.a == x) {
            return true;
        } else {
            for (int i = 0; i < this.b.length(); ++i) {
                if (this.b.charAt(i) == x) {
                    return true;
                }
            }

            return false;
        }
    }

    public boolean tieneEnI(char x) {
        for (int i = 0; i < this.b.length(); ++i) {
            if (this.b.charAt(i) == x) {
                return true;
            }
        }

        return false;
    }

    public boolean esMayuscula(char x) {
        return x > 64 && x < 91;
    }

    public boolean esMinuscula(char x) {
        return x > 96 && x < 123;
    }
}
```

Adiciones, Eliminaciones, Uniones e Intersecciones de elementos

```java
import java.util.Stack;

public class ProductionsSet {

    protected int n = 0;
    protected Production[] C = new Production[99];

    public ProductionsSet() {
    }

    public int nroElems() {
        return this.n;
    }

    public void mostrar() {
        System.out.print("{");

        for (int i = 0; i < this.n; ++i) {
            System.out.println(this.C[i].getImpresion() + ", ");
        }

        System.out.println("} " + this.nroElems());
    }

    public void adicionar(Production X) {
        for (int i = 0; i < this.n; ++i) {
            if (this.C[i].esIgual(X)) {
                return;
            }
        }

        this.C[this.n] = X;
        ++this.n;
    }
```

```java
    public void eliminar(int j) {
        for (int i = j; i < this.n; ++i) {
            this.C[i - 1] = this.C[i];
        }

        --this.n;
    }

    public String getImpresion() {
        String I = "";

        for (int i = 0; i < this.n; ++i) {
            I = I + "  " + this.C[i].getImpresion() + "\n";
        }

        return I;
    }

    public String[] getImpresionV() {
        String[] V = new String[this.n];

        for (int i = 0; i < this.n; ++i) {
            V[i] = this.C[i].getImpresion();
        }

        return V;
    }

    public Production getElemento(int i) {
        return this.C[i];
    }

    public ProductionsSet union(ProductionsSet B) {
        ProductionsSet U = new ProductionsSet();
```

```
        int i;
        for (i = 0; i < this.n; ++i) {
            U.adicionar(this.C[i]);
        }

        for (i = 0; i < B.n; ++i) {
            U.adicionar(B.C[i]);
        }

        return U;
    }

    public ProductionsSet interseccion(ProductionsSet B) {
        ProductionsSet I = new ProductionsSet();

        for (int i = 0; i < this.n; ++i) {
            for (int j = 0; j < B.n; ++j) {
                if (this.C[i].equals(B.C[j])) {
                    I.adicionar(this.C[i]);
                    break;
                }
            }
        }

        return I;
    }

    public boolean existeProduccion(char X, char Y) {
        for (int i = 0; i < this.nroElems(); ++i) {
            if (this.getElemento(i).getA() == X && this.getElemento(i).getB().charAt(0) == Y) {
                return true;
            }
        }

```

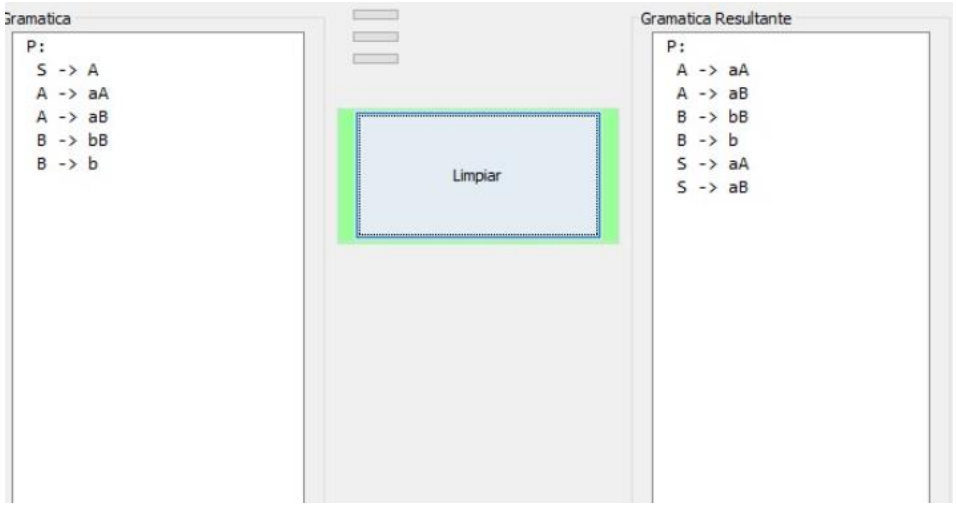
```
    public boolean existeProduccion(char X, char Y) {
        for (int i = 0; i < this.nroElems(); ++i) {
            if (this.getElemento(i).getA() == X && this.getElemento(i).getB().charAt(0) == Y) {
                return true;
            }
        }

        return false;
    }

    public boolean esGenerada(char X, char Y) {
        Stack pila = new Stack();
        pila.push(X + "");

        for (boolean sw = false; !pila.isEmpty(); sw = true) {
            char W = ((String) pila.pop()).charAt(0);
            if (W == Y) {
                return true;
            }

            if (W == X && sw) {
                return false;
            }

            for (int i = 0; i < this.nroElems(); ++i) {
                if (this.getElemento(i).getA() == W) {
                    pila.push(this.getElemento(i).getB());
                }
            }
        }

        return false;
    }
}
```

FUNCIONAMIENTO

Muerta o Inútil

Inaccesible

Redenominación



CONCLUSIONES

Dada la complejidad de la práctica concluimos que sin duda fue una práctica de gran nivel, es una práctica que nos hizo esforzarnos al 1000 por cierto, puesto que había muchas cosas que no dominábamos del lenguaje de programación java y debimos buscar, investigar e indagar para lograr sacar adelante el objetivo de la misma, por otro lado mencionar que tuvimos que cambiar al lenguaje Java porque se intentó en C pero se complicaba demasiado el hecho de controlar las funciones de re dominación y accesibilidad, dada la situación no nos dimos por vencidos y nos quedamos con un grato sabor de boca al lograr que nuestro programa saliera a flote sin problemas, al tener las ideas claras acerca del programa nos dimos cuenta que no existía opción y el segundo lenguaje que mejor dominábamos era Java. Por lo anterior, nos llenamos de gratitud con la profesora y la unidad de aprendizaje porque sin duda nos está convirtiendo en unos alumnos dominadores no solo de un lenguaje sino de 2 y claro lograr aún más lenguajes.

Se logro el planteamiento del problema con éxito y un buen trabajo en equipo.