

**Universidad De San Carlos De Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas**

**Compiladores 1
Aux. Erick Lemus
Sección “B”**

**REGEXIVE
Proyecto 1**

NOMBRE	CARNET
Jose Castro Sincu	201504115

Guatemala 09 de Marzo 2021

Manual Tecnico

El programa REGEXIVE cuenta con varias clase de las cuales tienes metodos, atributos.

El programa cuenta con tres paquetes las cuales son:

Analizadores

Arbol

Regexive

El paquete analizadores contiene las clases siguientes:

Lexico.java:

Esta clase posee todas los alfabetos y expresiones regulares y las palabras reservadas.

Lexico.jflex:

Reglas declaradas

Sintactico.cup:

Es este archivo se declararon todas las reglas del parser, como terminales, no terminales, y producciones. Metodos para graficar y variables globales.

```
terminal String BARINVERTIDA;
terminal String CORCERRADO;
terminal String POTENCIA;
terminal String GBAJO;
terminal String LLAVEIZQ;
terminal String OR;
terminal String LLAVEDER;
terminal String ALCANCE;

non terminal Nodo instrucciones;
non terminal Nodo bloque;
non terminal Nodo expresion_regula

non terminal Nodo letras_min;
non terminal Nodo letras_may;
non terminal Nodo lista_numeros;
non terminal Nodo simbolos;
//non terminal Nodo comprobar;
```

Sintactico.java:

Clase generada del archivo cup.

Generador_de_Archivos.java:

Atraves de esta clase se genero los archivos sintactico y lexico.

Sym.java:

Es la clase auxiliar de Sintatico.java

El paquete Arbol contiene las siguientes clases:

Anulabilidad.java:

En esta clase su objetivo esta calcular los anulables, primeros, ultimos, y el follow de la expresion regular.

Contiene los metodos de concatenacion, or, una vez, y una o mas veces.

```

L  /**
   package arbol;
   //import analizadores.*;

   import java.util.LinkedList;

   /**...4 lines */
   public class Anulabilidad {

       private int numera_hoja;
       public Anulabilidad()
       {...6 lines }

       public void llenar_datos_concatenacion(LinkedList<Siguietes> lista_siguietes, Nodo nuevaConcat, Nodo a, Nodo b)
       {...90 lines } //fin de llenar datos concatenacion

       public void llenar_datos_or(Nodo nuevaOr, Nodo a, Nodo b) {...52 lines } //fin del metodo llenar datos or

       public void llenar_datos_aste(LinkedList<Siguietes> lista_siguietes, Nodo nuevaPor, Nodo a)
       {...39 lines } //fin de llenar datos asterisco y interrogacion

       public void llenar_datos_interr(Nodo nuevaInter, Nodo a)
       {...19 lines }

       public void llenar_datos_mas(LinkedList<Siguietes> lista_siguietes, Nodo nuevaMas, Nodo a) {...49 lines }
   }

```

Conjunto_Simbolos.java:

Posee una lista en la cual se guardara todas las listas o caracteres que formaran parte de la expresion regular.

```

   package arbol;

   import java.util.LinkedList;

   /**...4 lines */
   public class Conjunto_Simbolos {
       private String id;
       private LinkedList<Integer> letra_conjunto;

       public Conjunto_Simbolos()
       {...4 lines }
       public Conjunto_Simbolos(String id, char inicio, char fin)
       {...5 lines }

       /**...3 lines */
       public String getId() {...3 lines }

       /**...3 lines */
       public void setId(String id) {...3 lines }

       /**...3 lines */
       public LinkedList<Integer> getLetra_conjunto() {...3 lines }

       /**...3 lines */
       public void setLetra_conjunto(LinkedList<Integer> letra_conjunto) {...3 lines }
   }

```

Dato_Column.java:

Su objetivo es ser encabezado de la tabla de transiciones en la cual posee tres atributos que son fila, columna y letra.

```
+ /**...4 lines */
public class DatoColumn {

+   /**...3 lines */
+   public int getFila() {...3 lines }

+   /**...3 lines */
+   public void setFila(int fila) {...3 lines }

+   /**...3 lines */
+   public int getColumna() {...3 lines }

+   /**...3 lines */
+   public void setColumna(int Columna) {...3 lines }

+   /**...3 lines */
+   public String getLetra() {...3 lines }

+   /**...3 lines */
+   public void setLetra(String letra) {...3 lines }
  private int fila;
  private int columna;
  private String letra;

  public DatoColumn(int fila,int columna,String identificador)
+   {...6 lines }
}
```

Estados.java:

Es una clase en la cual se crea una lista de estados. Para la cual su funciones es no conteners estados repetid.

```

    * @param estadoAceptacion the estadoAceptacion to set
    */
    public void setEstadoAceptacion(String estadoAceptacion) {...3 lines }

    /**...3 lines */
    public String getNombre_estado() {...3 lines }

    /**...3 lines */
    public void setNombre_estado(String nombre_estado) {...3 lines }

    /**...3 lines */
    public LinkedList<String> getEstado() {...3 lines }

    /**...3 lines */
    public void setEstado(LinkedList<String> estado) {...3 lines }

    public void addEstado_nuevo(String valor)
    {...3 lines }
    private String nombre_estado;
    private LinkedList<String> estado;
    private String nombre_hoja;
    private String estadoAceptacion;

    public Estados(String nombre_es,String nombr_hoj, String estaAcep)
    {...7 lines }

    public Estados(String nombre_estad,LinkedList<String> estado,String nombr_hoj, String estaAcep)
    {
        this.nombre_estado = nombre_estad;
    }

```

Exp_Reg.java:

Contiene las reglas necerasias. Para evaluar los textos de entrada.

Graficas.java:

Sirve para hacer las graficas de las de los afd, afn, etc.

Nodo.java:

Esta clase sirve para crear los arboles. Los cuales se genera la expresion.

```

public class Nodo {

    private int numera_hoja;
    private Nodo hizq;
    private Nodo hder;

    private String valor;
    private int id;

    private int id_hoja;
    private String anulable;
    private LinkedList<String> primeros;
    private LinkedList<String> ultimos;
    private LinkedList<String> siguientes;

    public Nodo(Nodo hizq, Nodo hder, String valor, int id) {
        this.numera_hoja = 0;
        this.hizq = hizq;
        this.hder = hder;
        this.valor = valor;
        this.id = id;

        this.id_hoja = 0;
        this.anulable = "";
        this.primeros = new LinkedList<String>();
        this.ultimos = new LinkedList<String>();
        this.siguientes = new LinkedList<String>();
    }
}

```

Siguientes.java:

Contienes una lista de las cuales salen los siguientes de las hojas.

Tabla_Conjunto.java:

Thompson.java:

Se crea la grafica del metodo de thompson.

Transiciones.java:

En esta clase se crea, la tabla de transiciones y la grafica del afn.

```
private LinkedList<DatoColum> temp_encabezado;
private LinkedList<Estados> lista_estado;
private int contar_estado;
private int cont_fila;
private int fila ;
private int columna;

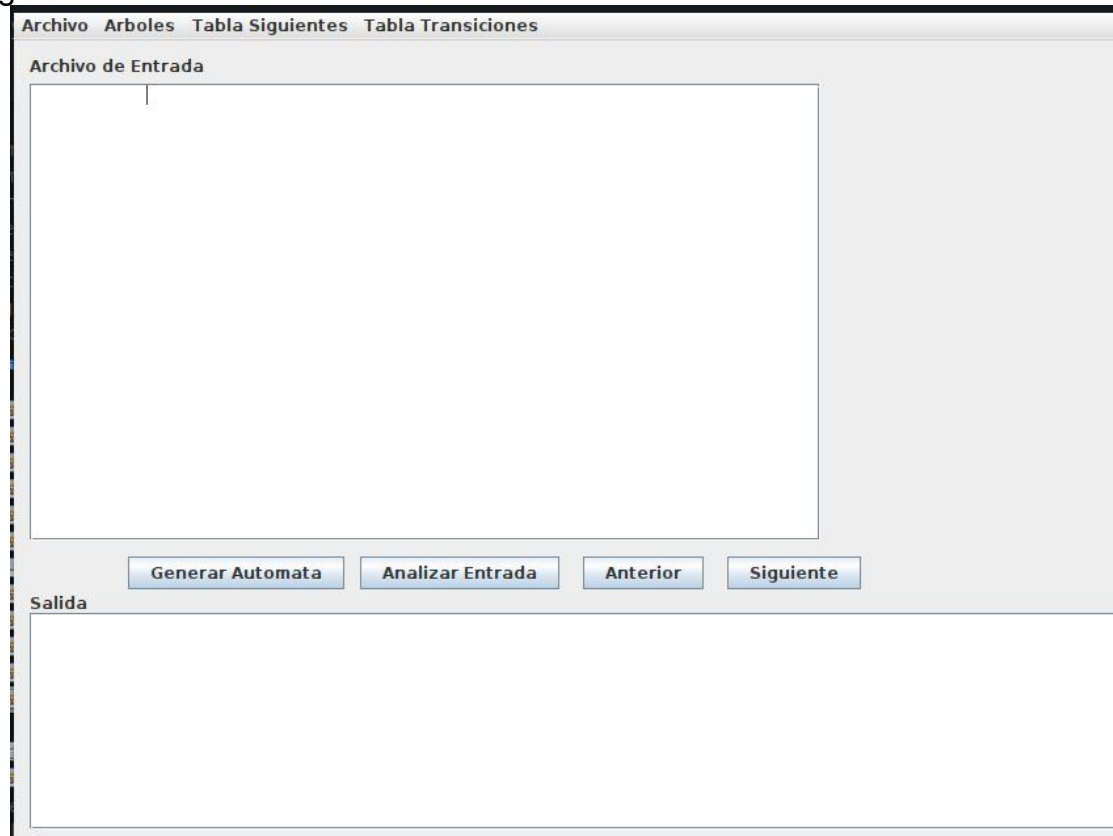
public Transiciones()
{
    this.lista_siguientes = new LinkedList<>();
    this.temp_encabezado = new LinkedList<>();
    this.lista_estado = new LinkedList<>();
    this.temp_encabezado.add(new DatoColum(0,0,"Estado"));
    contar_estado=0;
    cont_fila = 1;
    fila = 0;
    columna =0;
}

public void tabla_transiciones(LinkedList<Siguientes> lista_sig,LinkedList<String> primeros){
    //le asigno el valor de los siguientes a la nueva lista
    lista_siguientes = lista_sig;
    int temp_colum = 1;
```

El paquete Regexive contiene las siguientes clase:

Grafica.java:

Contiene todas las funcionalidades del programa como boton analizar, cargar archivos. Etc.



Regexive.java:

Es el programa principal que llama la interfaz grafica.

```
5  //
6  package regexive;
7  import arbol.*;
8  import java.util.LinkedList;
9
10 /**
11  *
12  * @author dark
13  */
14 public class Regexive {
15
16     /**
17     * @param args the command line arguments
18     */
19     public static void main(String[] args) {
20         // TODO code application logic here
21
22
23         Grafica graf= null;
24         graf = new Grafica();
25         graf.show();
26         System.out.println("Hola mundo");
27     }
28 }
29
30 }
```