

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

ESTRUCTURA DE DATOS

SECCION B

PROYECTO 1

Nombre: Kemel Josué Efraín Ruano Jeronimo

Carné: 202006373

Fecha de entrega: 19/09/2022

OBJETIVOS

- Aprender usar herramienta jflex y cup.
- Aplicar conocimiento de compiladores para las traducciones.
- Comprender el funcionamiento de gramáticas tipo LR1, LSR.
- Aplicar conocimiento sobre análisis sintáctico y léxico .

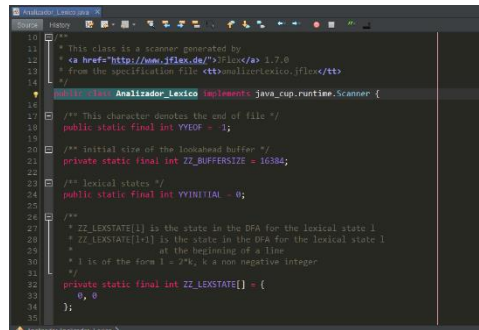
INFORMACION DEL SISTEMA

Requisitos del sistema:

- Windows 10,8,7 (x86 y x64).
- Procesador a 1.6 GHz o superior.
- 1 GB (32 bits) o 2 GB (64 bits) de RAM (agregue 512 MB al host si se ejecuta en una máquina virtual).
- 3 GB de espacio disponible en el disco duro.
- Disco duro de 5400 RPM.
- Tarjeta de vídeo compatible con DirectX 9 con resolución de pantalla de 1024 x 768 o más.

CLASES UTILIZADAS

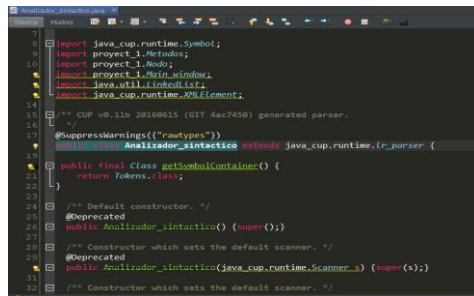
- **public class Analizador_Lexico**



```
10: /**
11:  * This class is a scanner generated by
12:  * http://www.jflex.de/ 1.7.0
13:  * from the specification file <tt>analizadorlexico.jflex</tt>
14:  */
15:
16: public class Analizador_Lexico implements java_cup.runtime.Scanner {
17:
18:     /** This character denotes the end of file */
19:     public static final int YEOF = 1;
20:
21:     /** Initial size of the lookahead buffer */
22:     private static final int ZZ_BUFFERSIZE = 16384;
23:
24:     /** lexical states */
25:     public static final int YYINITIAL = 0;
26:
27:     /**
28:      * ZZ_LKSTATE[] is the state in the DFA for the lexical state 1
29:      * ZZ_LKSTATE[1+] is the state in the DFA for the lexical state 1
30:      * at the beginning of a line
31:      * 1 is of the form 1 + 2^k, k a non-negative integer
32:      */
33:     private static final int ZZ_LKSTATE[] = {
34:         0, 0
35:     };
36: }
```

Esta clase es generada por jflex la cual contiene los estados de cada token que es aceptado por el sistema el cual luego genera una lista de tokens que la pasa al sintáctico

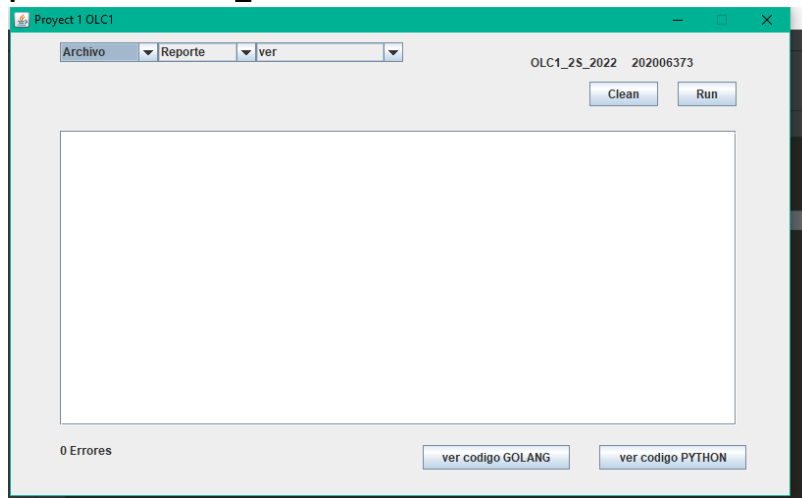
- **public class Analizador_sintactico**



```
7:
8: import java_cup.runtime.Symbol;
9: import project_1.Metodos;
10: import project_1.Metodos;
11: import project_1.Metodos;
12: import java.util.LinkedList;
13: import java_cup.runtime.Symbol;
14:
15: /** CUP v0.11b 20160615 (GIT dac7450) generated parser.
16:  */
17: @SuppressWarnings({"rawtypes"})
18: public class Analizador_sintactico extends java_cup.runtime.lr_parser {
19:
20:     public final Class getSymbolContainer() {
21:         return Tokens.class;
22:     }
23:
24:     /** Default constructor. */
25:     @Deprecated
26:     public Analizador_sintactico() {super();}
27:
28:     /** Constructor which sets the default scanner. */
29:     @Deprecated
30:     public Analizador_sintactico(java_cup.runtime.Scanner s) {super(s);}
31:
32:     /** Constructor which sets the default scanner. */
33: }
```

Esta clase es generada por cup la cual hace nuestro análisis sintactico para lo que se necesita una gramática ya sea recursiva por la izquierda o derecha.

- **public class Main_window**



Esta clase genera un ventana interactiva con el usuario el cual puede traducir dos lenguajes asi como ver los reportes .

- **public class Metodos**

```

1 package project_1;
2
3
4 import java.io.File;
5 import java.io.FileNotFoundException;
6 import java.io.FileWriter;
7 import java.io.IOException;
8 import javax.swing.JOptionPane;
9 import Analizador.Analizador_Lexico;
10 import java.io.PrintWriter;
11 public class Metodos {
12
13
14
15     public static String Validados(String Tipo,String valor,int CountValor){
16         String SalidaGolang = "";
17         boolean numero = (valor.matches("[0-9]+"));
18         boolean decimal = (valor.matches("[0-9]+[.][0-9]+"));
19         String T_po = Tipo.toLowerCase();
20         String T_po2 = valor.toLowerCase();
21         boolean is_NAL = (valor.matches(".*[{}].*"));
22         boolean is_NAL2 = (valor.matches("[']([\\w]+[']|'')"));
23         if(numero == true){
24             SalidaGolang += " "+ "int" + " "+ "="+ " ";
25             for(int j = 1; j <= CountValor; ++j) {
26                 SalidaGolang += " "+ valor;

```

Esta clase es la encargada de traducir la mayor parte de código al lenguaje de golang .