

Analizador JSON

```
ini ::= instrucciones;  
  
instrucciones ::= instruccion instrucciones  
| instruccion  
;  
  
instruccion ::=  
    LLI declaraciones:a LLC;  
  
declaraciones ::= declaracion:a declaraciones:b  
|  
declaracion:a  
;  
  
declaracion ::=  
    COMD CADENA:a COMD DOSPT DECIMAL:b  
| COMD CADENA:a COMD DOSPT COMD CADENA:b COMD  
| COMD CADENA:a COMD DOSPT DECIMAL:b COMA  
| COMD CADENA:a COMD DOSPT COMD CADENA:b COMD COMA  
;  
  
CADENA ::=  
    identificador:a  
| DECIMAL:a  
| DECIMAL:a CADENA:b  
| identificador:a CADENA:b  
;
```

Analizador StatPy

```
package AnalizadorStatPy;
import java_cup.runtime.*;
import statpyconverter.Config;
import statpyconverter.Variable;
import statpyconverter.Token;
import java.util.ArrayList;
import java.io.PrintWriter;
import java.io.IOException;

//PARSER
parser code
{:

    Config cf = new Config();
    ArrayList<Variable> arr = new ArrayList<>();
    ArrayList<Token> toks = new ArrayList<>();
    String tituloGB = "";
    String tituloGBX = "";
    String tituloGBY = "";
    String[] ejeXGB;
    String[] valoresGB;
    String tituloGP = "";
    String[] ejeXGP;
    String[] valoresGP;

    public void guardarGlobales(String tipo, String nombre, String dato) {
        Variable varr = new Variable(tipo, nombre, dato);
        arr.add(varr);
        //System.out.println(varr.getNombre()+" -> "+varr.getDato());
    }

    public String buscarGlobales(String nombre){
        for(int i =0; i<arr.size();i++){
            if(arr.get(i).getNombre().equals(nombre)){
                return arr.get(i).getDato();
            }
        }
        return "";
    }

    public void listaTokens(String lexema,String token,String linea,String
columna){
        Token tk = new Token(lexema,token,linea,columna);
```

```

        toks.add(tk);
        //System.out.println("lexema:"+lexema+" token:"+token+"
linea:"+linea+" columna:"+columna);
    }

    public void htmlTokens(ArrayList<Token> tk){
        try {
            try (FileWriter fileWriter = new
FileWriter("C:\\Users\\1998j\\OneDrive\\Desktop\\compil\\proyecto1\\StatPyCo
nverter\\src\\Reportes\\tokens.html")) {
                fileWriter.write("<h1>Lista de tokens</h1>");
                for(int i = 0; i<tk.size();i++){
                    fileWriter.write("<p>Token: "+tk.get(i).getToken()+ "
Lexema:"+tk.get(i).getLexema()+
                    " Línea:"+tk.get(i).getLinea()+
Columna:"+tk.get(i).getColumna()+"</p><br>");
                }
            }
        } catch (IOException e) {
            System.out.println("Error al escribir en el archivo: " +
e.getMessage());
        }
    }

    public void syntax_error(Symbol s){
        System.out.println("Error R de sintaxis: "+ s.value +" Línea
"+(s.left+1)+" columna "+(s.right+1) );
    }

    public void unrecovered_syntax_error(Symbol s) throws
java.lang.Exception{
        System.out.println("Error NR de sintaxis: "+ s.value +" Línea
"+(s.left+1)+" columna "+(s.right+1) );
    }

:}

//TERMINALES
terminal String
PTCOMA, PAR_IZQ, PAR_DER, COMS, COMD, DOSPT, LLI, LLC, DOLLAR, COMA, CORI, CORD, PT;
terminal String MAS, MENOS, POR, DIV, UMENOS;
terminal String ENTERO, DECIMAL;
terminal String varINT, varDOUBLE, varCHAR, varBOOL, varSTRING;
terminal String
wVoid, wMain, wPrint, wIF, wELSE, wSWITCH, wCASE, wBREAK, wFOR, wWHILE, wDO, wNewValor;

```

```
terminal String wDefGlob,wGraphB,wGraphP;  
terminal String wTITULO,wEJEX,wVALORES,wTITULOX,wTITULOY;  
terminal String RELACIONES,IGUAL;  
terminal String AND,OR,NOT;  
terminal String identificador;
```

```
//NO_TERMINALES
```

```
non terminal ini;  
non terminal instrucciones;  
non terminal instruccion;  
non terminal expresion;  
non terminal declaraciones;  
non terminal declaracion;  
non terminal declaracionINT;  
non terminal declaracionDOUBLE;  
non terminal declaracionCHAR;  
non terminal declaracionBOOL;  
non terminal declaracionSTRING;  
non terminal declaracionPRINT;  
non terminal main;  
non terminal sentenciaIF;  
non terminal sentenciaELSE;  
non terminal sentenciaSWITCH;  
non terminal sentenciaCASE;  
non terminal sentenciaFOR;  
non terminal sentenciaWHILE;  
non terminal sentenciaDOWHILE;  
non terminal CONDICIONES;  
non terminal CASE;  
non terminal CADENA;  
non terminal declaracionFUNCIONES;  
non terminal declaracionesG;  
non terminal declaracionG;  
non terminal declaracionINTG;  
non terminal declaracionDOUBLEG;  
non terminal declaracionCHARG;  
non terminal declaracionBOOLG;  
non terminal declaracionSTRINGG;  
non terminal declaracionesGB;  
non terminal decTituloGB;  
non terminal decTituloGBX;  
non terminal decTituloGBY;  
non terminal decEjeX;  
non terminal chainX;  
non terminal decValores;
```

```

non terminal declaracionesGP;
non terminal declaracionGB;
non terminal declaracionGP;
non terminal decTituloGP;
non terminal decEjeXP;
non terminal decValoresP;

//PRECEDENCIA_DE_OPERADORES
precedence left MAS,MENOS;
precedence left POR,DIV;
precedence right UMENOS;
precedence left AND;
precedence left OR;
precedence right NOT;

start with ini;

ini ::= instrucciones
;

instrucciones ::= instruccion:a instrucciones:b
| instruccion:a
;

instruccion ::=
    main:a
;

main ::=
    wVoid wMain PAR_IZQ PAR_DER LLI declaraciones:a
LLC
;

declaraciones ::= declaracion:a declaraciones:b
| declaracion:a
;

declaracion ::=
    declaracionINT:a
    | declaracionDOUBLE:a
    | declaracionCHAR:a
    | declaracionBOOL:a
    | declaracionSTRING:a
    | declaracionPRINT:a

```

```

| sentenciaIF:a
| sentenciaELSE:a
| sentenciaSWITCH:a
| sentenciaFOR:a
| sentenciaWHILE:a
| sentenciaDOWHILE:a
| declaracionFUNCIONES

sentenciaIF ::=
    wIF PAR_IZQ CONDICIONES:a PAR_DER LLI declaraciones:b LLC
;

sentenciaELSE ::=
    wELSE sentenciaIF:a
;

sentenciaSWITCH ::=
    wSWITCH PAR_IZQ identificador:a PAR_DER LLI sentenciaCASE:b
LLC
;

sentenciaCASE ::=
    sentenciaCASE:a CASE:b
    | CASE:a
;

CASE ::=
    wCASE CADENA:a DOSPT declaraciones:b wBREAK
PTCOMA
;

sentenciaFOR ::=
    wFOR PAR_IZQ declaracion:a identificador:b RELACIONES:c CADENA:d PTCOMA
identificador:f MAS MAS PAR_DER LLI declaraciones:g LLC
;

sentenciaWHILE ::=
    wWHILE PAR_IZQ identificador:a RELACIONES:b CADENA:c PAR_DER LLI
declaraciones:d LLC
;

sentenciaDOWHILE ::=
    wDO LLI declaraciones:a LLC wWHILE PAR_IZQ identificador:b RELACIONES:c
expresion:d PAR_DER PTCOMA
;

```

```

declaracionINT ::=
    varINT identificador:a IGUAL expresion:b PTCOMA
    | varINT identificador:a
PTCOMA
;

declaracionDOUBLE ::=
    varDOUBLE identificador:a IGUAL expresion:b PTCOMA
    | varDOUBLE identificador:a
PTCOMA
;

declaracionCHAR ::=
    varCHAR identificador:a IGUAL COMS identificador:b COMS
PTCOMA
    | varCHAR identificador:a IGUAL COMD identificador:b COMD PTCOMA
    | varCHAR identificador:a PTCOMA
;

declaracionBOOL ::=
    varBOOL identificador:a IGUAL identificador:b PTCOMA
;

declaracionSTRING ::=
    varSTRING identificador:a IGUAL COMS CADENA:b COMS
PTCOMA
    | varSTRING identificador:a IGUAL COMD CADENA:b COMD
PTCOMA
    | varSTRING identificador:a
PTCOMA
;

declaracionPRINT ::=
    wPrint PAR_IZQ expresion:a PAR_DER PTCOMA
    | wPrint PAR_IZQ COMS CADENA:a COMS MAS identificador:b PAR_DER
PTCOMA
    | wPrint PAR_IZQ COMS CADENA:a DOSPT COMS MAS identificador:b PAR_DER
PTCOMA
    | wPrint PAR_IZQ COMS CADENA:a COMS PAR_DER
PTCOMA
    | wPrint PAR_IZQ COMD CADENA:a COMD MAS identificador:b PAR_DER
PTCOMA
    | wPrint PAR_IZQ COMD CADENA:a DOSPT COMD MAS identificador:b PAR_DER
PTCOMA

```

```

    | wPrint PAR_IZQ COMD CADENA:a COMD PAR_DER
PTCOMA
;

expresion ::=
    MENOS expresion:a
    | expresion:a MAS      expresion:b
    | expresion:a MENOS    expresion:b
    | expresion:a
POR      expresion:b
    | expresion:a
DIV      expresion:b
    | ENTERO:a

    | DECIMAL:a

    | PAR_IZQ expresion:a
PAR_DER
    | DOLLAR LLI wNewValor COMA COMD CADENA:a COMD COMA COMD
identificador:b COMD LLC
;

CADENA ::=
    identificador:a

    |
ENTERO:a
    | ENTERO:a
CADENA:b
    | identificador:a
CADENA:b
    | identificador:a PT CADENA:b
;

CONDICIONES ::=
    identificador:a RELACIONES:b
CADENA:c
    | identificador:a RELACIONES:b CADENA:c AND
CONDICIONES:d
    | identificador:a RELACIONES:b CADENA:c OR
CONDICIONES:d

```



```

    | NOT
identificador:a
    | NOT identificador:a RELACIONES:b
CADENA:c
    | NOT identificador:a RELACIONES:b identificador:c
CONDICIONES:d
    | NOT identificador:a RELACIONES:b ENTERO:c
CONDICIONES:d
;

//FUNCIONES_ESTADISTICAS-----
-----

declaracionFUNCIONES::=
    wVoid wDefGlob PAR_IZQ PAR_DER LLI declaracionesG LLC
    | wVoid wGraphB PAR_IZQ PAR_DER LLI declaracionesGB LLC
    | wVoid wGraphP PAR_IZQ PAR_DER LLI declaracionesGP LLC
    | wVoid identificador:a PAR_IZQ PAR_DER LLI declaraciones:b LLC
;

declaracionesG::=
    declaracionG:a declaracionesG:b
    | declaracionG:a
;

declaracionG::=
    declaracionINTG
    | declaracionDOUBLEG
    | declaracionCHARG
    | declaracionBOOLG
    | declaracionSTRINGG
;

declaracionINTG::=
    varINT identificador:a IGUAL expresion:b PTCOMA
    | varINT identificador:a PTCOMA
;

declaracionDOUBLEG::=
    varDOUBLE identificador:a IGUAL expresion:b PTCOMA
    | varDOUBLE identificador:a
PTCOMA
;

```

```

declaracionCHARG::=
    varCHAR identificador:a IGUAL COMS identificador:b COMS PTCOMA
    | varCHAR identificador:a IGUAL COMD identificador:b COMD PTCOMA
    | varCHAR identificador:a PTCOMA
;

declaracionBOOLG::=
    varBOOL identificador:a IGUAL identificador:b PTCOMA
;

declaracionSTRINGG::=
    varSTRING identificador:a IGUAL COMS CADENA:b COMS PTCOMA
    | varSTRING identificador:a IGUAL COMD CADENA:b COMD PTCOMA
    | varSTRING identificador:a PTCOMA
;

declaracionesGB::=
    declaracionGB:a declaracionesGB:b
    | declaracionGB:a
;

declaracionGB::=
    decTituloGB
    | decTituloGBX
    | decTituloGBY
    | decEjeX
    | decValores
;

declaracionesGP::=
    declaracionGP:a declaracionesGP:b
    | declaracionGP:a
;

declaracionGP::=
    decTituloGP
    | decEjeXP
    | decValoresP
;

decTituloGB::=
    varSTRING wTITULO IGUAL identificador:a PTCOMA

```

```

    | varSTRING wTITULO IGUAL COMD identificador:a COMD PTCOMA
;

decTituloGBX::=
    varSTRING wTITULOX IGUAL identificador:a PTCOMA
    | varSTRING wTITULOX IGUAL COMD identificador:a COMD PTCOMA
;

decTituloGBY::=
    varSTRING wTITULOY IGUAL identificador:a PTCOMA
    | varSTRING wTITULOY IGUAL COMD identificador:a COMD PTCOMA
;

decEjeX::=
    varSTRING CORI CORD wEJEX IGUAL LLI chainX:a LLC PTCOMA
;

chainX::=
    COMD CADENA:a COMD COMA chainX:b
    | CADENA:a COMA chainX:b
    | CADENA:a
    | DOLLAR LLI wNewValor COMA COMD CADENA:a COMD COMA COMD identificador:b
COMD LLC
    | COMA chainX:a
;

decValores::=
    varDOUBLE CORI CORD wVALORES IGUAL LLI chainX:a LLC PTCOMA
;

decTituloGP::=
    varSTRING wTITULO IGUAL identificador:a PTCOMA
    | varSTRING wTITULO IGUAL COMD identificador:a COMD PTCOMA
;

decEjeXP::=
    varSTRING CORI CORD wEJEX IGUAL LLI chainX:a LLC PTCOMA
;

decValoresP::=
    varDOUBLE CORI CORD wVALORES IGUAL LLI chainX:a LLC PTCOMA
;

```