UNIVERSIDAD TECNOLÓGICA DE SANTIAGO, UTESA
SISTEMA CORPORATIVO

FACULTAD DE INGENIERÍA Y ARQUITECTURA
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

ASIGNACIÓN:
COMPILADORES
INF-920-001

TAREA

Actividad. Semana 4

PARA
PROF. IVAN MENDOZA

POR
ANTHONY TINEO CABREJA 1-19-0423

HECHO EN
SANTIAGO DE LOS CABALLEROS, REPÚBLICA DOMINICANA
18 DE JUNIO DEL 2024

Crear un analizador Sintáctico (Cualquier tipo, LL, LR o sin usar estos métodos) en el Lenguaje de programación de su preferencia.

1. Ejercicio. Crear un analizador Sintáctico (Cualquier tipo, LL, LR o sin usar estos métodos) en el Lenguaje de programación de su preferencia.

- Netbeans 8.2
- Java JDK 8 2.

2. Documentar Lenguaje utilizado del analizador Léxico para realizar pruebas.

```
package codigo;

import java_cup.runtime.Symbol;

%% /*Declaraciones que vamos a utilizar*/

%class LexerCup

%type java_cup.runtime.Symbol

%cup /*Realizara el analisis*/

%full /*Permite recordar toda la cadena*/

%line /*Nos regresa la linea*/

%char /*Nos regresa la columna en la que se encuentra*/

L=[a-zA-Z_]+

D=[0-9]+

espacio=[ ,\t,\r,\n]+ /*Los espacios que ignorara el analizador lexico*/

%{

    private Symbol symbol(int type, Object value){

        return new Symbol(type, yyline, yycolumn, value);

    }

    private Symbol symbol(int type){

        return new Symbol(type, yyline, yycolumn);

    }

%}

%%
```

```
/*Palabras reservadas*/

/* Espacios en blanco */

{espacio} {/*Ignore*/}



/* Comentarios */

( "//"(.)* ) {/*Ignore*/}



/* Comillas */

( "\"" ) {return new Symbol(sym.Comillas, yychar, yyline, yytext());}



/* Tipos de datos */

( byte | char | long | float | double ) {return new Symbol(sym.T_dato,
yychar, yyline, yytext());}



/* Tipo de dato Int (Para el main) */

( "int" ) {return new Symbol(sym.Int, yychar, yyline, yytext());}



/* Tipo de dato String */

( String ) {return new Symbol(sym.Cadena, yychar, yyline, yytext());}



/* Palabra reservada If */

( if ) {return new Symbol(sym.If, yychar, yyline, yytext());}



/* Palabra reservada Else */

( else ) {return new Symbol(sym.Else, yychar, yyline, yytext());}



/* Palabra reservada Do */

( do ) {return new Symbol(sym.Do, yychar, yyline, yytext());}
```

```
/* Palabra reservada While */

( while ) {return new Symbol(sym.While, yychar, yyline, yytext());}


/* Palabra reservada For */

( for ) {return new Symbol(sym.For, yychar, yyline, yytext());}


/* Operador Igual */

( "=" ) {return new Symbol(sym.Igual, yychar, yyline, yytext());}


/* Operador Suma */

( "+" ) {return new Symbol(sym.Suma, yychar, yyline, yytext());}


/* Operador Resta */

( "-" ) {return new Symbol(sym.Resta, yychar, yyline, yytext());}


/* Operador Multiplicacion */

( "*" ) {return new Symbol(sym.Multiplicacion, yychar, yyline, yytext());}


/* Operador Division */

( "/" ) {return new Symbol(sym.Division, yychar, yyline, yytext());}


/* Operadores logicos */

( "&&" | "||" | "!" | "&" | "|" ) {return new Symbol(sym.Op_logico, yychar,
yyline, yytext());}


/*Operadores Relacionales */
```

```
( ">" | "<" | "==" | "!=" | ">=" | "<=" | "<<" | ">>" ) {return new
Symbol(sym.Op_relacional, yychar, yyline, yytext());}



/* Operadores Atribucion */

( "+=" | "-="  | "*=" | "/=" | "%=" | "=" ) {return new
Symbol(sym.Op_atribucion, yychar, yyline, yytext());}



/* Operadores Incremento y decremento */

( "++" | "--" ) {return new Symbol(sym.Op_incremento, yychar, yyline,
yytext());}



/*Operadores Booleanos*/

( true | false ) {return new Symbol(sym.Op_booleano, yychar, yyline,
yytext());}



/* Parentesis de apertura */

( "(" ) {return new Symbol(sym.Parentesis_a, yychar, yyline, yytext());}



/* Parentesis de cierre */

( ")" ) {return new Symbol(sym.Parentesis_c, yychar, yyline, yytext());}



/* Llave de apertura */

( "{" ) {return new Symbol(sym.Llave_a, yychar, yyline, yytext());}



/* Llave de cierre */

( "}" ) {return new Symbol(sym.Llave_c, yychar, yyline, yytext());}



/* Corchete de apertura */

( "[" ) {return new Symbol(sym.Corchete_a, yychar, yyline, yytext());}
```

```
/* Corchete de cierre */

( "]" ) {return new Symbol(sym.Corchete_c, yychar, yyline, yytext());}



/* Marcador de inicio de algoritmo */

( "main" ) {return new Symbol(sym.Main, yychar, yyline, yytext());}



/* Punto y coma */

( ";" ) {return new Symbol(sym.P_coma, yychar, yyline, yytext());}



/* Identificador */

{L}({L}|{D})* {return new Symbol(sym.Identificador, yychar, yyline,
yytext());}



/* Numero */

("(-"{D}+")")|{D}+ {return new Symbol(sym.Numero, yychar, yyline,
yytext());}



/* Error de analisis */

. {return new Symbol(sym.ERROR, yychar, yyline, yytext());}

```

4. Imagen de previsualización.