

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Curso: Laboratorio Organización de Lenguajes & Compiladores 1
Auxiliar: José Diego Pérez Toralla
Auxiliar: Maynor Octavio Piló Tuy



Juan F. Urbina S. **2019060651**
Sección: **C**

Guatemala, Marzo de 2023

Índice

Léxico	3
Sintáctico	6
Terminales	6
No Terminales.....	6
Gramática	7

Léxico

```
espacios_muchos = [ \t\r\n\f]+
//letras & digitos
letra = [a-zA-Z]
digito = [0-9]
numero = {digito}+
s_flecha = -(\\s)*>
//comentarios
comentario_simple = "//" [^"\\n"]*
comentario_multiple = "<!" [^"!>"]* "!">"
llave_abierta = "{"
llave_cerrada = "}"
dos_puntos = ":"
punto_coma = ";"
s_porcentaje = "%%"
virgilla = "~"
coma = ","
punto = "."
or = "|"
asterisco = "*"
s_mas = "+"
s_interrogacion = "?"
fin_linea = "\\n"
s_comilla = "\\\""
doble_comilla = "\\\""
range = [!-/] | [:-@] | [\\[-` | [\\{-\\}]
espacio = "\\ \"'"
conj_sym = ["c"|"C"]["o"|"O"]["n"|"N"]["j"|"J"]
ident = {letra}({letra}|{digito}|"_")*
string_dat = [^"\\\\"]* "\\\""
s_frase = "\"" {string_dat}* [^"\\"]* {string_dat}* "\""

```

<YYINITIAL>{

{letra} { tokens tmp = new tokens(yytext(), yyline, yycolumn); TokensList.add(tmp); return new Symbol(sym.letra, yycolumn, yyline, yytext()); }

{numero} { tokens tmp = new tokens(yytext(), yyline, yycolumn); TokensList.add(tmp); return new Symbol(sym.numero, yycolumn, yyline, yytext()); }

{comentario_simple} { /*System.out.println("Comentario Simple: "+ yytext());*/ tokens tmp = new tokens(yytext(), yyline, yycolumn); TokensList.add(tmp); }

{comentario_multiple} { /*System.out.println("Comentario Multiple: "+ yytext());*/ tokens tmp = new tokens(yytext(), yyline, yycolumn); TokensList.add(tmp); }

{llave_abierta} { tokens tmp = new tokens(yytext(), yyline, yycolumn); TokensList.add(tmp); return new Symbol(sym.llave_abierta, yycolumn, yyline, yytext()); }

{llave_cerrada} { tokens tmp = new tokens(yytext(), yyline, yycolumn); TokensList.add(tmp); return new Symbol(sym.llave_cerrada, yycolumn, yyline, yytext()); }

{dos_puntos} { tokens tmp = new tokens(yytext(), yyline, yycolumn); TokensList.add(tmp); return new Symbol(sym.dos_puntos, yycolumn, yyline, yytext()); }

{punto_coma} { tokens tmp= new tokens(yytext(), yyline, yycolumn); TokensList.add(tmp); return new Symbol(sym.punto_coma, yycolumn, yyline, yytext()); }

{s_flecha} { tokens tmp = new tokens(yytext(), yyline, yycolumn); TokensList.add(tmp); return new Symbol(sym.s_flecha, yycolumn, yyline, yytext()); }

{s_porcentaje} { tokens tmp = new tokens(yytext(), yyline, yycolumn); TokensList.add(tmp); return new Symbol(sym.s_porcentaje, yycolumn, yyline, yytext()); }

{virgilla} { tokens tmp = new tokens(yytext(), yyline, yycolumn); TokensList.add(tmp); return new Symbol(sym.virgilla, yycolumn, yyline, yytext()); }

{coma} { tokens tmp = new tokens(yytext(), yyline, yycolumn); TokensList.add(tmp); return new Symbol(sym.coma, yycolumn, yyline, yytext()); }

{punto} { tokens tmp = new tokens(yytext(), yyline, yycolumn); TokensList.add(tmp); return new Symbol(sym.punto, yycolumn, yyline, yytext()); }

{or} { tokens tmp = new tokens(yytext(), yyline, yycolumn); TokensList.add(tmp); return new Symbol(sym.or, yycolumn, yyline, yytext()); }

{asterisco} { return new Symbol(sym.asterisco, yycolumn, yyline, yytext()); }

{s_mas} { tokens tmp = new tokens(yytext(), yyline, yycolumn); TokensList.add(tmp); return new Symbol(sym.s_mas, yycolumn, yyline, yytext()); }

{s_interrogacion} { tokens tmp = new tokens(yytext(), yyline, yycolumn); TokensList.add(tmp); return new Symbol(sym.s_interrogacion, yycolumn, yyline, yytext()); }

{fin_linea} { tokens tmp = new tokens(yytext(), yyline, yycolumn); TokensList.add(tmp); return new Symbol(sym.fin_linea, yycolumn, yyline, yytext()); }

{s_comilla} { tokens tmp = new tokens(yytext(), yyline, yycolumn); TokensList.add(tmp); return new Symbol(sym.s_comilla, yycolumn, yyline, yytext()); }

```

    {doble_comilla}      { tokens tmp = new tokens(yytext(), yyline, yycolumn); TokensList.add(tmp); return new
Symbol(sym.doble_comilla, yycolumn, yyline, yytext()); }

    {range}             { tokens tmp = new tokens(yytext(), yyline, yycolumn); TokensList.add(tmp); return new
Symbol(sym.range, yycolumn, yyline, yytext()); }

    {espacio}           { tokens tmp = new tokens(yytext(), yyline, yycolumn); TokensList.add(tmp); return new
Symbol(sym.espacio, yycolumn, yyline, yytext()); }

    {conj_sym}          { tokens tmp = new tokens(yytext(), yyline, yycolumn); TokensList.add(tmp); return new
Symbol(sym.conj_sym, yycolumn, yyline, yytext()); }

    {ident}             { tokens tmp = new tokens(yytext(), yyline, yycolumn); TokensList.add(tmp); return new
Symbol(sym.ident, yycolumn, yyline, yytext()); }

    {s_frase}           { tokens tmp = new tokens(yytext(), yyline, yycolumn); TokensList.add(tmp); return new
Symbol(sym.s_frase, yycolumn, yyline, yytext()); }

}

{espacios_muchos} { }

```

Sintáctico

Terminales

terminal numero;
terminal llave_abierta,llave_cerrada;
terminal dos_puntos,punto_coma,s_flecha;
terminal s_porcentaje,virgilla,coma,punto;
terminal or,asterisco,s_mas,s_interrogacion;
terminal fin_linea,s_comilla,doble_comilla;
terminal range,espacio,conj_sym,ident,s_frase,letra;

No Terminales

non terminal INICIO;
non terminal STARTS;
non terminal CONTENIDO;
non terminal CONJUNTO;
non terminal CONTENIDOR;
non terminal ER;
non terminal DEFCONJ;
non terminal SEPCOMAS;
non terminal RANGO;
non terminal DATOSEP;
non terminal SEPCOMASR;
non terminal DATORANGO;
non terminal DEFER;
non terminal OP;
non terminal REFCONJ;
non terminal CADENAS;
non terminal CADENASR;
non terminal NAMECOBJ;

Gramática

start with INICIO;

STARTS ::= llave_abierta CONTENIDO

;

CONTENIDO ::= CONJUNTO CONTENIDOR

| ER CONTENIDOR

;

CONJUNTO ::= conj_sym dos_puntos NAMECOBJ:a s_flecha DEFCONJ

;

DEFCONJ ::= SEPCOMAS

| RANGO

;

SEPCOMAS ::= DATOSEP:a SEPCOMASR

;

DATOSEP ::= numero:a

| letra:a

| range:a

| asterisco:a

| s_mas:a

| coma:a

| punto:a

| dos_puntos:a

| punto_coma:a

| s_interrogacion:a

| llave_abierta:a

| or:a

| llave_cerrada:a

| doble_comilla:a

| s_comilla:a

| fin_linea:a

| s_frase:a

;

SEPCOMASR ::= coma SEPCOMAS

| punto_coma

;

RANGO ::= DATORANGO:a virgilla DATORANGO:b punto_coma ;

DATORANGO ::= numero:a

| letra:a

| range:a

| espacio:a

| asterisco:a

| s_mas:a

| coma:a

| punto:a

| dos_puntos:a

| punto_coma:a

| s_interrogacion:a

| llave_abierta:a

| or:a

| llave_cerrada:a

| doble_comilla:a

| s_comilla:a

| fin_linea:a

;

ER ::= ident:a s_flecha DEFER ;

DEFER ::= OP:a

| REFCONJ:a

| s_frase:a DEFER

| espacio:a DEFER

| fin_linea:a DEFER

| s_comilla:a DEFER

| doble_comilla:a DEFER

| punto_coma:a

;


```

OP ::= or:a DEFER
    | asterisco:a DEFER
    | s_mas:a DEFER
    | s_interrogacion:a DEFER
    | punto:a DEFER
;

REFCONJ ::= llave_abierta NAMECOBJ:a llave_cerrada DEFER
;

NAMECOBJ ::= ident:a
    | letra:a
;

CONTENIDOR ::= s_porcentaje CADENASR
    | CONTENIDO
;

CADENAS ::= s_porcentaje CADENASR
    | ident:a dos_puntos s_frase:b punto_coma CADENASR
;

CADENASR ::= llave_cerrada
    | CADENAS
;

```