M Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas
Organización Lenguajes y Compiladores 1

Nombre: Selvin Lisandro Aragón Pérez
Carnet: 201701133

# MANUAL TECNICO

## ANALISADOR LEXICO

### EXPRESION REGULAR

Id | Logicos | Relacionales | Digito | Simbolos | Cadena | CadHTML | Comentario | ComentarioMul

**Donde:**

**Cc:** Cualquier Cosa

**L:** Letras

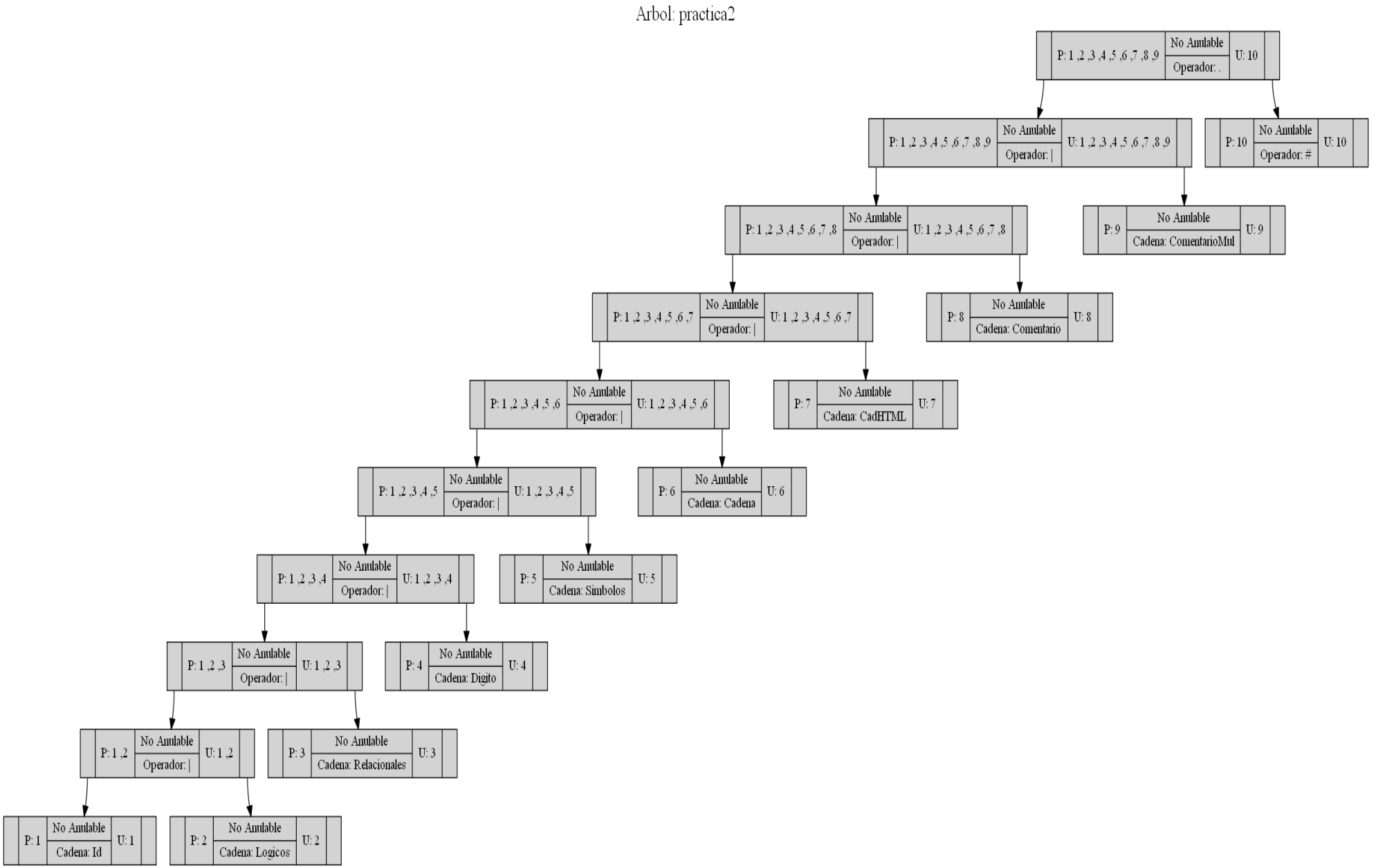
**D:** Digitos

**S:** Simbolos

### TABLA DE TOKENS

| No. | Símbolo | Descripción |
|---|---|---|
| 1 | //* | Comentario |
| 2 | <!!> | Comentario Multilínea |
| 3 | Res | Reservada |
| 4 | Id | Identificador |
| 5 | “Cc” | Cadena |
| 6-21 | Símbolos | Símbolos |
| 22 | D | Digito |
| 23 | S_Logico | Lógico |
| 24 | S_Relacionales | Relacionales |
| 25 | ‘C_Html’ | Cadena HTML |

**ARBOL**


Arbol: practica2
P: 1 ,2 ,3 ,4 ,5 ,6 ,7 ,8 ,9 | No Anulable | Operador: . | U: 10
P: 1 ,2 ,3 ,4 ,5 ,6 ,7 ,8 ,9 | No Anulable | Operador: | | U: 1 ,2 ,3 ,4 ,5 ,6 ,7 ,8 ,9
P: 10 | No Anulable | Operador: # | U: 10
P: 1 ,2 ,3 ,4 ,5 ,6 ,7 ,8 | No Anulable | Operador: | | U: 1 ,2 ,3 ,4 ,5 ,6 ,7 ,8
P: 9 | No Anulable | Cadena: ComentarioMul | U: 9
P: 1 ,2 ,3 ,4 ,5 ,6 ,7 | No Anulable | Operador: | | U: 1 ,2 ,3 ,4 ,5 ,6 ,7
P: 8 | No Anulable | Cadena: Comentario | U: 8
P: 1 ,2 ,3 ,4 ,5 ,6 | No Anulable | Operador: | | U: 1 ,2 ,3 ,4 ,5 ,6
P: 7 | No Anulable | Cadena: CadHTML | U: 7
P: 1 ,2 ,3 ,4 ,5 | No Anulable | Operador: | | U: 1 ,2 ,3 ,4 ,5
P: 6 | No Anulable | Cadena: Cadena | U: 6
P: 1 ,2 ,3 ,4 | No Anulable | Operador: | | U: 1 ,2 ,3 ,4
P: 5 | No Anulable | Cadena: Simbolos | U: 5
P: 1 ,2 ,3 | No Anulable | Operador: | | U: 1 ,2 ,3
P: 4 | No Anulable | Cadena: Digito | U: 4
P: 1 ,2 | No Anulable | Operador: | | U: 1 ,2
P: 3 | No Anulable | Cadena: Relacionales | U: 3
P: 1 | No Anulable | Cadena: Id | U: 1
P: 2 | No Anulable | Cadena: Logicos | U: 2

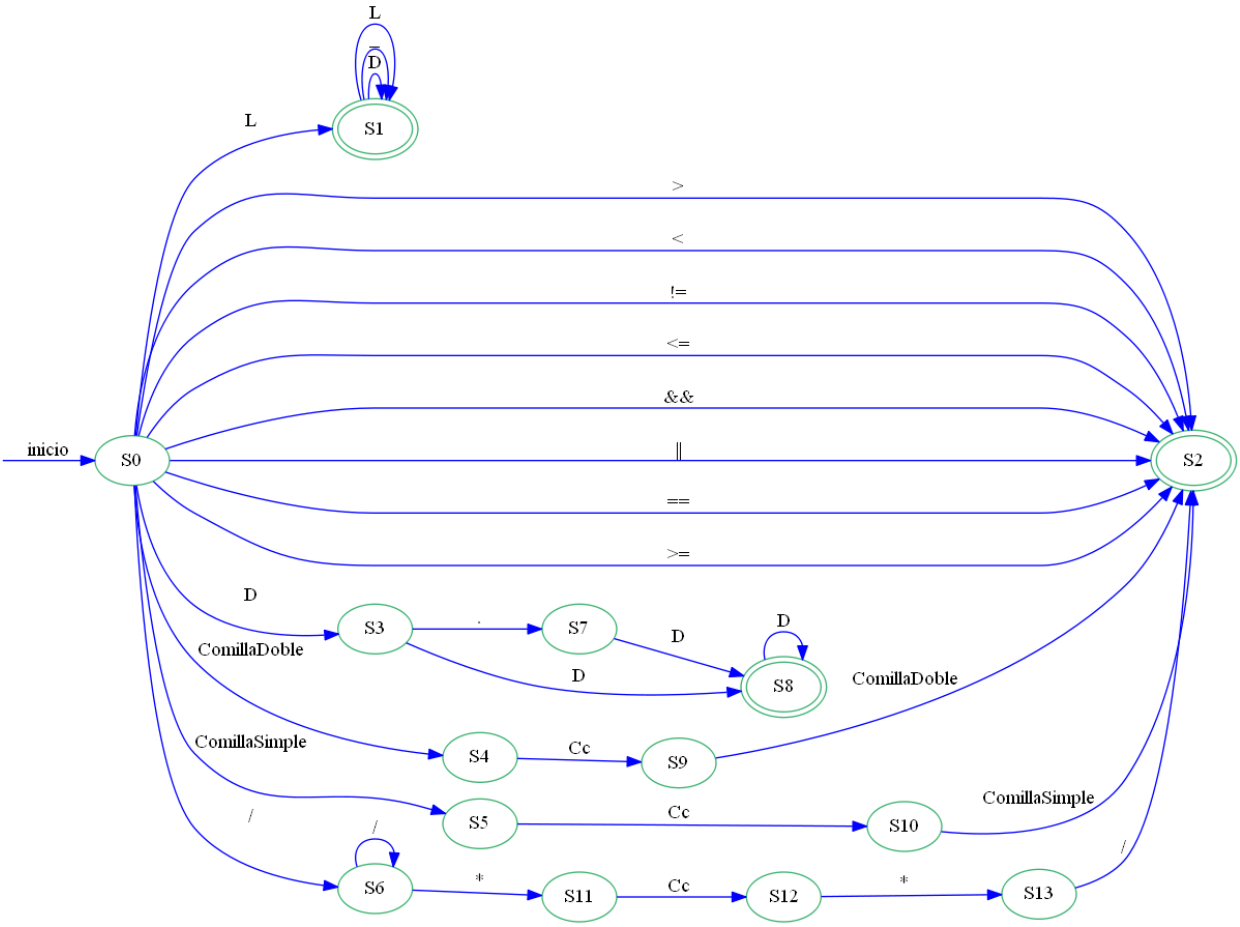
## TABLA DE SIGUIENTES

| Nombre de Hoja | Id de Hoja | Siguientes |
|---|---|---|
| "L" | 1 | 2,3,4,27 |
| "L" | 2 | 2,3,4,27 |
| "D" | 3 | 2,3,4,27 |
| "_" | 4 | 2,3,4,27 |
| "&&" | 5 | 27 |
| "\|\|" | 6 | 27 |
| ">" | 7 | 27 |
| "<" | 8 | 27 |
| ">=" | 9 | 27 |
| "<=" | 10 | 27 |
| "==" | 11 | 27 |
| "!=" | 12 | 27 |
| "D" | 13 | 13,14,15 |
| "." | 14 | 15 |
| "D" | 15 | 15,27 |
| "ComillaDoble" | 16 | 17 |
| "Cc" | 17 | 18 |
| "ComillaDoble" | 18 | 27 |
| "ComillaSimple" | 19 | 20 |
| "Cc" | 20 | 21 |
| "ComillaSimple" | 21 | 27 |
| "/" | 22 | 22,23 |

| | | |
|---|---|---|
| "*" | 23 | 24 |
| "Cc" | 24 | 25 |
| "*" | 25 | 26 |
| "/" | 26 | 27 |
| # | 27 | |

## AFD

AFD: practica2



# ANALISADOR SINTACTICO

<INICIO> -> void <MAIN_METODO>
| <FUNCIONES>

<MAIN_METODO> -> main () { <SENTENCIAS> }
| id ( <PARAMETROS> ) { <SENTENCIAS_M> }

<PARAMETROS> -> <DECLARACION>
| epsilon

<SENTENCIAS> -> if ( <CONDICION> ){ <SENTENCIAS> } <ELSE><SENTENCIAS>
| switch ( Digito_Id ) { <CASOS> }<SENTENCIAS>
| for ( <DECLARACION> ; <CONDICION> ; <ITERADOR> ){ <SENTENCIAS_CICLO> }<SENTENCIAS>
| while ( <CONDICION> ){ <SENTENCIAS_CICLO> }<SENTENCIAS>
| do { <SENTENCIAS_CICLO> } while ( <CONDICION> ); <SENTENCIAS>
| Console.WriteLine( <CADENA_IMPRIMIR> ); <SENTENCIAS>
| <DECLARACION><SENTENCIAS>
| epsilon

<ELSE> -> else <IF_O_NO>
|epsilon

<IF_O_NO> -> { <SENTENCIAS> }
| if ( <CONDICION> ){ <SENTENCIAS> } <ELSE>

<CASOS> -> Case Valor : <SENTENCIAS_CASE> <CASOS>
| default : <SENTENCIAS_CASE> <CASOS>
| epsilon

<SENTENCIAS_CASE> -> <SENTENCIAS> <SENTENCIAS_CASE>
| break;
| épsilon

<SENTENCIAS_CICLO> -> <SENTENCIAS> <SENTENCIAS_CICLO>
| break;
| continue;
| épsilon

<SENTENCIAS_M> -> <SENTENCIAS> <SENTENCIAS_M>
| return;
| épsilon

<FUNCIONES> -> <TIPO> id ( <PARAMETROS> ){ <SENTENCIAS_F> }

<SENTENCIAS_F> -> <SENTENCIAS> <SENTENCIAS_F>
| return <EXPRESION> ;
| épsilon

<DECLARACION> -> <TIPO> <L_ID> <DECLARACION_TIPO>

<DECLARACION_TIPO> -> ;
| = <EXPRESION> ;

<L_ID> -> id <L_ID>
|, id <L_ID>
|epsilon

<EXPRESION> -> Digito <OPERACION>
| id -> o id()
| “ Cadena ” <OPERACION>

<OPERACION> -> Aritmetico Digito<OPERACION>
| + “ Cadena ” <OPERACION>
| epsilon

<CONDICION> -> Digito Relacionales Digito <MAS_COND>

<MAS_COND> -> Logicos <CONDICION>
| epsilon

## OTROS

- IDE: Visual Studio Code
- Lenguaje: JavaScript -> TypeScript
- Librerías Utilizadas:
    - Graphviz
    - NodeJs
    - Bootstrap
    - Css