



MANUAL TECNICO

ANALISADOR LEXICO

EXPRESION REGULAR

Id | Logicos | Relacionales | Dígito | Símbolos | Cadena | CadHTML | Comentario | ComentarioMul

Donde:

Cc: Cualquier Cosa

L: Letras

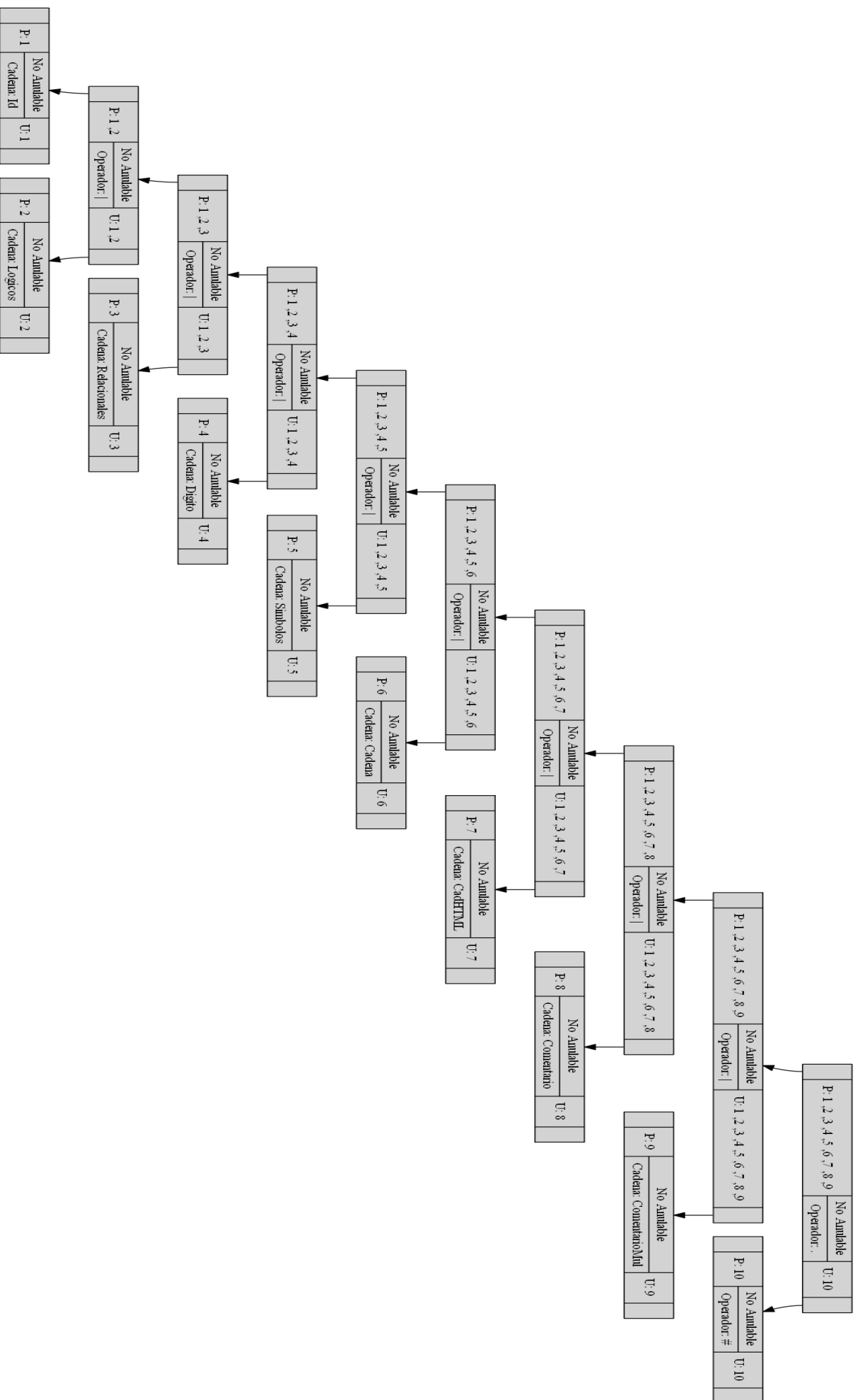
D: Dígitos

S: Símbolos

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	Dígito
23	S_Logico	Lógico
24	S_Relacionales	Relacionales
25	‘C_Html’	Cadena HTML

Arbol: practica2



ARBOL

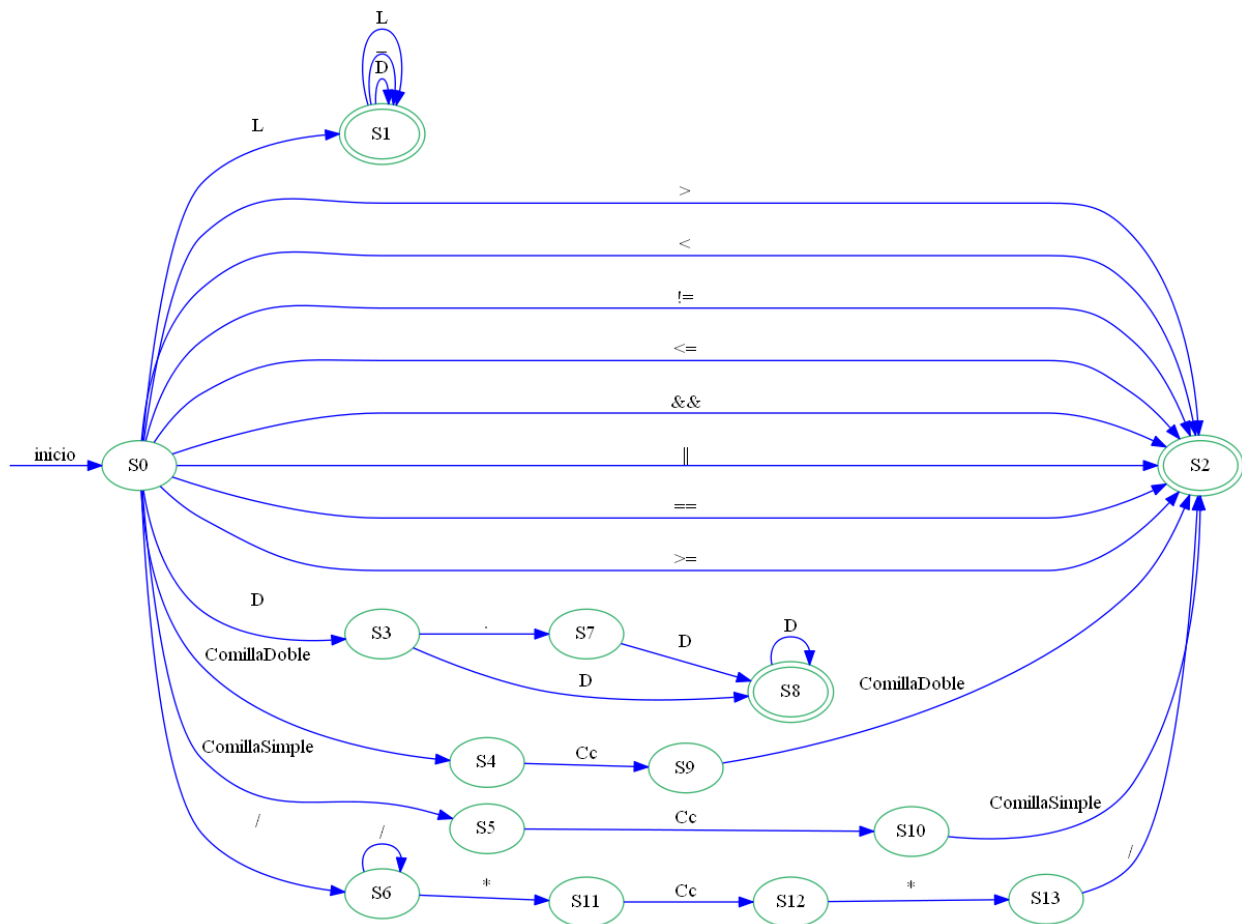
TABLA DE SIGUIENTES

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> -> / <COMENTARIO_TIPO>
          | void <MAIN_METODO>
          | <FUNCIONES>

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

<PARAMETROS> -> <DECLARACION>
                | epsilon

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

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

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

<COMENTARIO_TIPO> -> / <TODO> \n
                   | * <TODO_MUL> */

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

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

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

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

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

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

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

<OPERACION> -> Aritmético Dígito
| + “ Cadena ”
| epsilon

<CONDICION> -> Dígito Relacionales Dígito <MAS_COND>

<MAS_COND> -> Lógicos <CONDICION>
| epsilon

OTROS

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