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

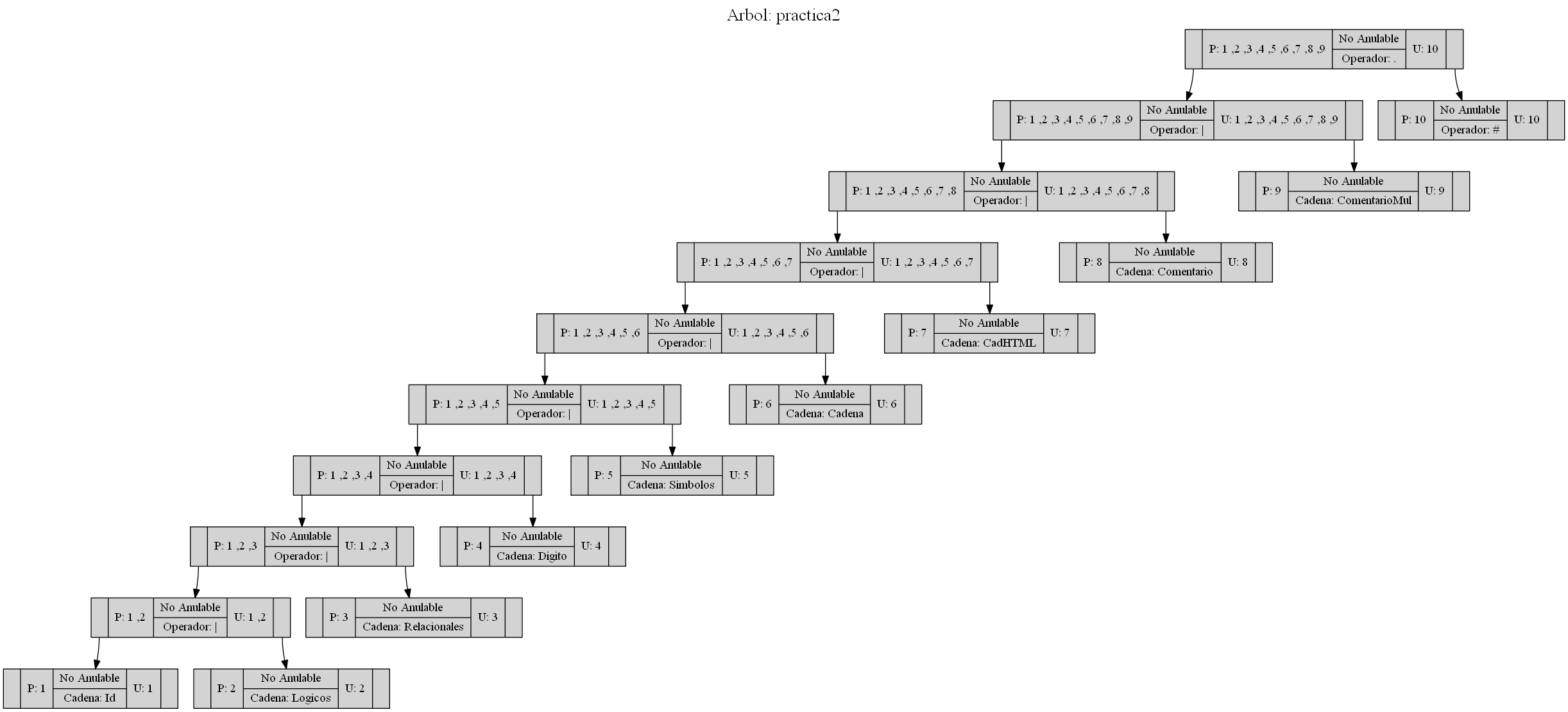
**D:** Digitos

**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 |

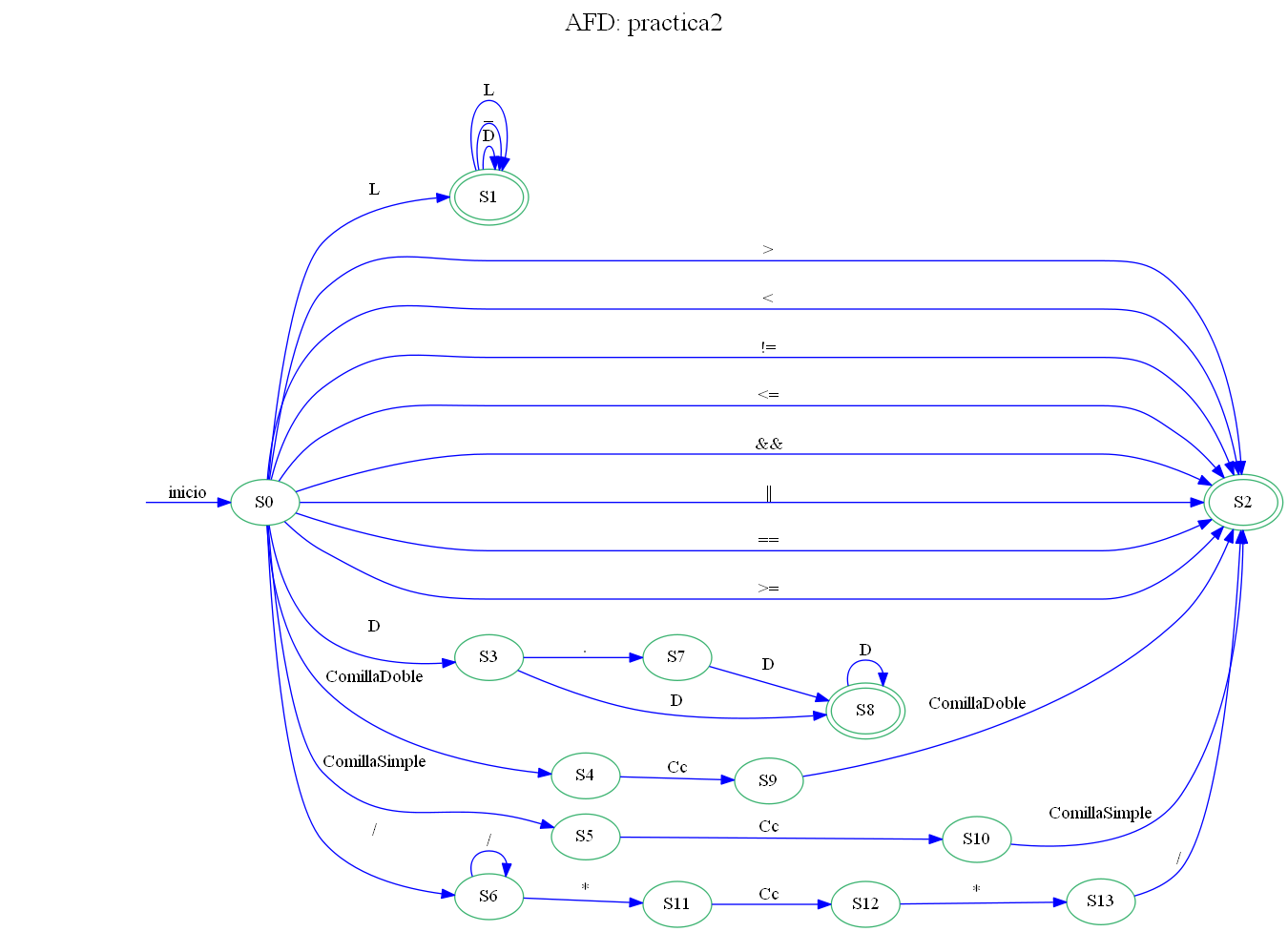
**ARBOL**



**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**



**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