MANUAL TÉCNICO

ANALIZADOR LÉXICO GCIC

TOKEN	LEXEMA
CGCIC	`[cC]"_"[gG][cC][iI][cC]`
CHEAD	`[cC]"_"[hH][eE][aA][dD]`
CTITLE	`[cC]"_"[tT][iI][tT][lL][eE]`
CLINK	`[cC]"_"[lL][iI][nN][kK]`
CBODY	`[cC]"_"[bB][oO][dD][yY]`
CSPAM	`[cC]"_"[sS][pP][aA][mM]`
CINPUT	`[cC]"_"[iI][nN][pP][uU][tT]`
CTEXT	`[cC]"_"[tT][eE][xX][tT][aA][rR][eE][aA]`
CSELECT	`[cC]"_"[sS][eE][lL][eE][cC][tT]`
COPTION	`[cC]"_"[oO][pP][tT][iI][oO][nN]`
CDIV	`[cC]"_"[dD][iI][vV]`
CIMG	`[cC]"_"[iI][mM][gG]`
CBR	`[cC]"_"[bB][rR]`
CBUTTON	`[cC]"_"[bB][uU][tT][tT][oO][nN]`
CH1	`[cC]"_"[hH]"1"`
CP	`[cC]"_"[pP]`
CSCRIPTING	`[cC]"_"[sS][cC][rR][iI][pP][tT][iI][nN][gG]`
STRING	`\"[^\"]*\"`

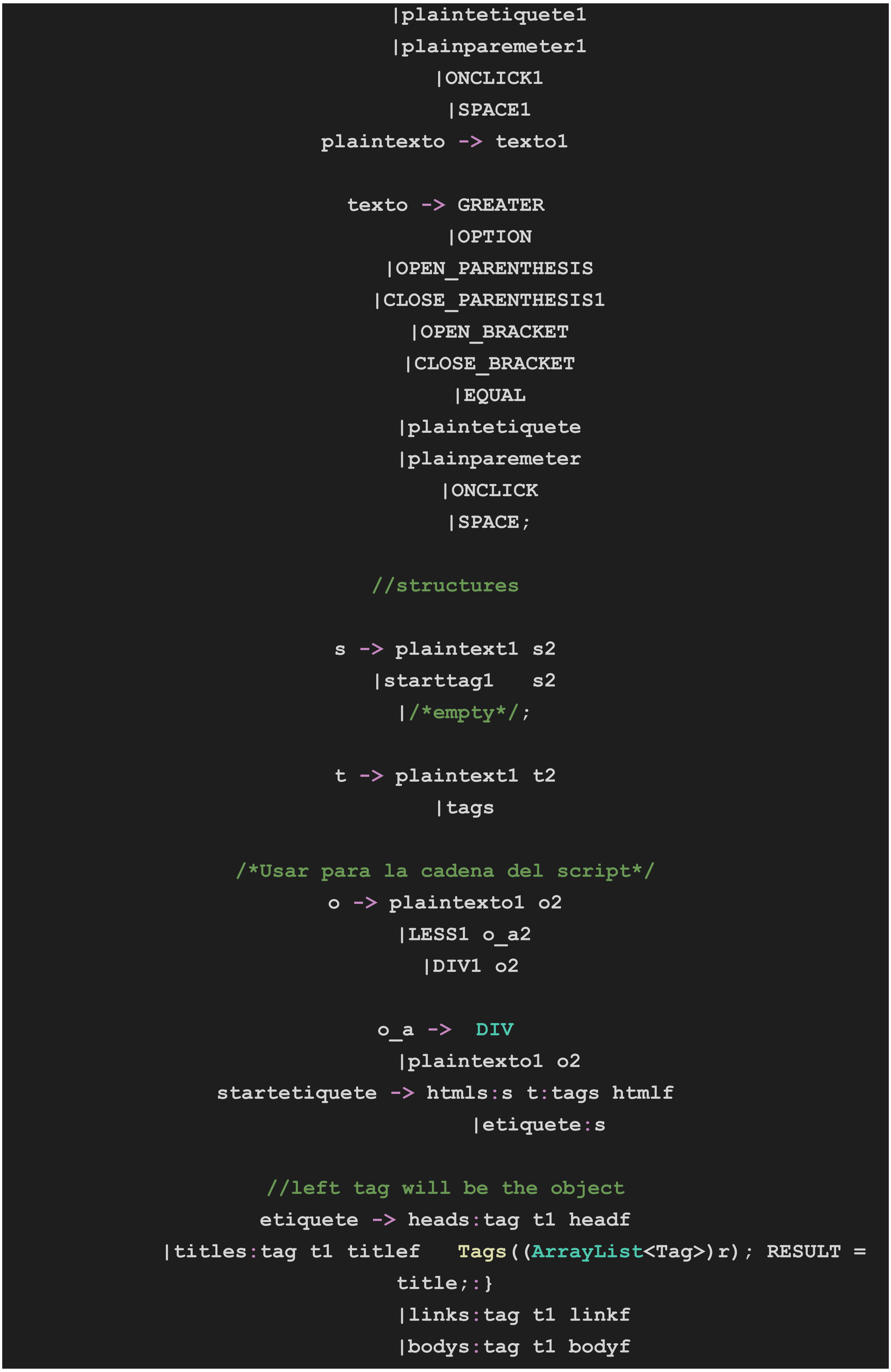
<table>
<tr><td></td><td></td></tr>
<tr><td>COMMENTARY</td><td>"!!"[^\n]+</td></tr>
<tr><td>BLOCK_COMMENTARY</td><td><!--([^-]|-[^-]|--+[^->])*-*--></td></tr>
<tr><td>DECIMAL</td><td>(([1-9][0-9]*|0?)[.])[0-9]+</td></tr>
<tr><td>NUMBER</td><td>[0-9]+</td></tr>
<tr><td>SYMCHAR</td><td>\'[^\']*\'</td></tr>
<tr><td>IdVariable</td><td>[aA-zZ][[aA-zZ]|[0-9]|[_]]*</td></tr>
<tr><td>PROCESS</td><td>[P][R][O][C][E][S][S][_][^ \\(\\\)\n\t\r]+</td></tr>
<tr><td>REDIRECT</td><td>“REDIRECT”</td></tr>
<tr><td>INIT</td><td>“INIT”</td></tr>
<tr><td>END</td><td>“END”</td></tr>
<tr><td>IF</td><td>IF</td></tr>
<tr><td>THEN</td><td>THEN</td></tr>
<tr><td>ELSE</td><td>ELSE</td></tr>
<tr><td>REPEAT</td><td>REPEAT</td></tr>
<tr><td>HUNTIL</td><td>HUNTIL</td></tr>
<tr><td>INSERT</td><td>INSERT</td></tr>
<tr><td>WHILE</td><td>THENWHILE</td></tr>
<tr><td>ON_LOAD</td><td>ON_LOAD</td></tr>
<tr><td>TRUE</td><td>true</td></tr>
<tr><td>FALSE</td><td>false</td></tr>
<tr><td>GLOBAL</td><td>@global</td></tr>
<tr><td>OR</td><td>||</td></tr>
<tr><td>AND</td><td>&&</td></tr>
</table>

NOT	!
EQUALIZATION	==
DIFFERENTIATION	!=
LESSTHAN	<=
GREATERTHAN	>=
LESS	<
GREATER	>
SUMA	+
RESTA	-
DIV	/
MULT	*
ASC	ASC
DESC	DESC
LETPAR_NUM	LETPAR_NUM
LETIMPAR_NUM	LETIMPAR_NUM
REVERSE	REVERSE
RANDCHAR	CARACTER_ALEATORIO
RANDINT	NUM_ALEATORIO
EXIT	EXIT
INT	integer
DECIMAL	decimal
BOOLEAN	boolean
CHAR	char
STRINGVAR	string
EQUAL	=

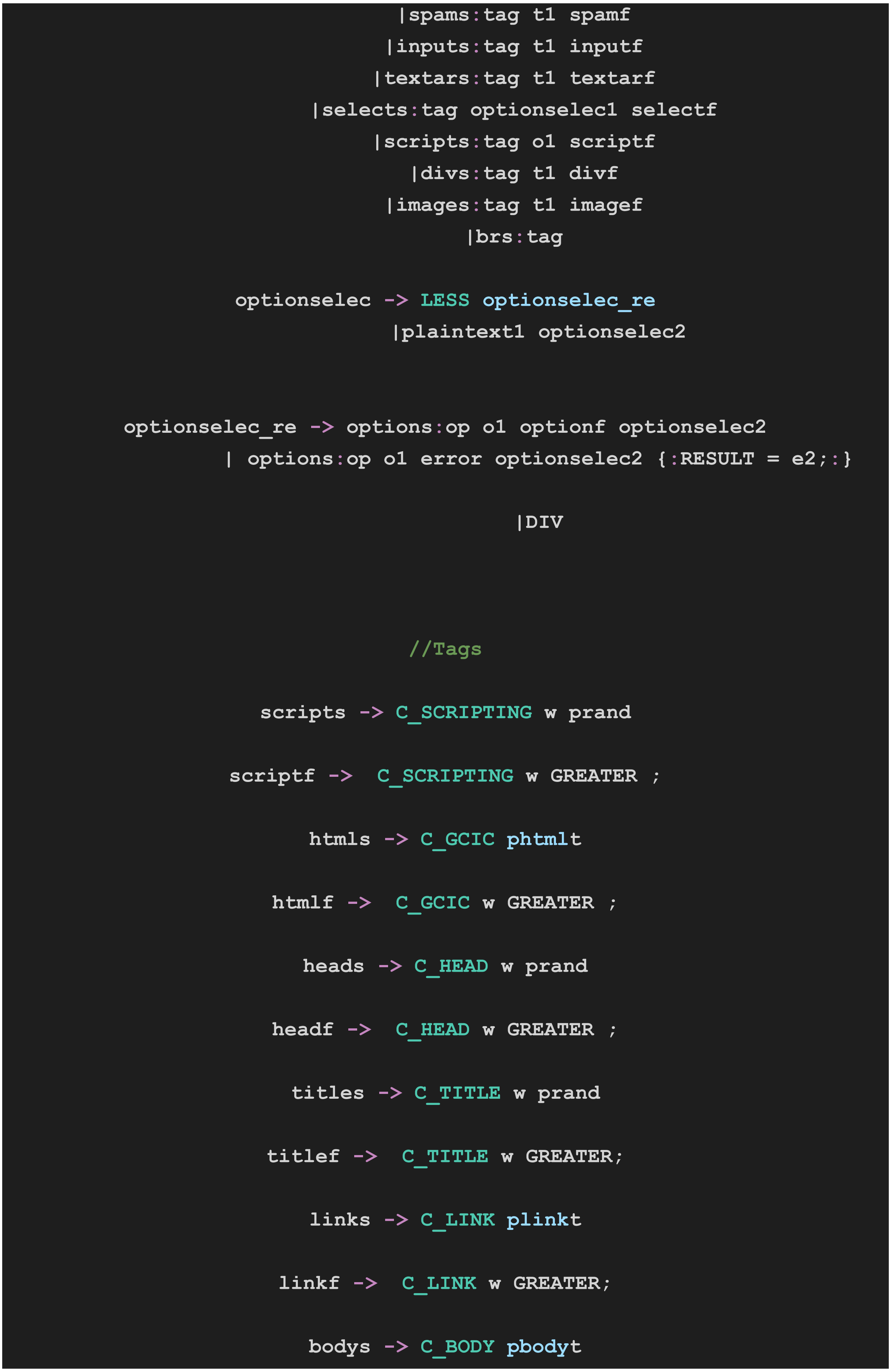
COLON	;
SEMICOLON	:
OPEN_CURLY	{
CLOSE_CURLY	}
OPEN_BRACKET	[
CLOSE_BRACKET	]
OPEN_PARENTHESIS	(
CLOSE_PARENTHESIS	)
GETELEMENT	getElementById

HTML ANALIZADOR SINTÁCTICO

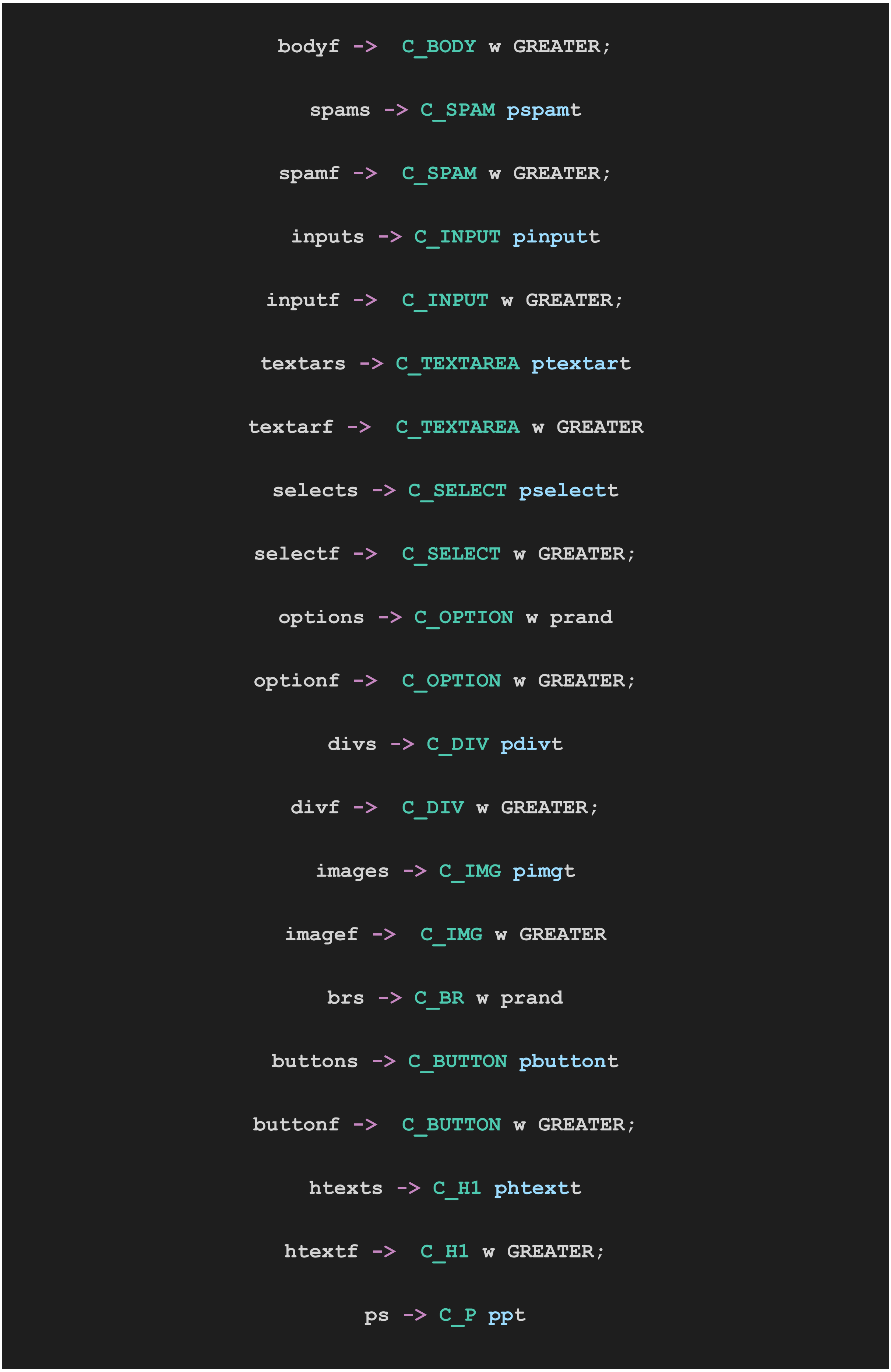
```
           //White spaces text
               w ->SPACE w
                 |/*empty*/;

               //plaintext

           tags -> LESS tags_re

         tags_re -> etiquetet tl
                          |DIV

     starttag -> LESS startetiquete

           plaintext -> text1

             text -> GREATER1
                        |DIV1
                      |OPTION1
                 |OPEN_PARENTHESIS1
                |CLOSE_PARENTHESIS1
                   |OPEN_BRACKET1
                  |CLOSE_BRACKET1
                       |EQUAL1
```



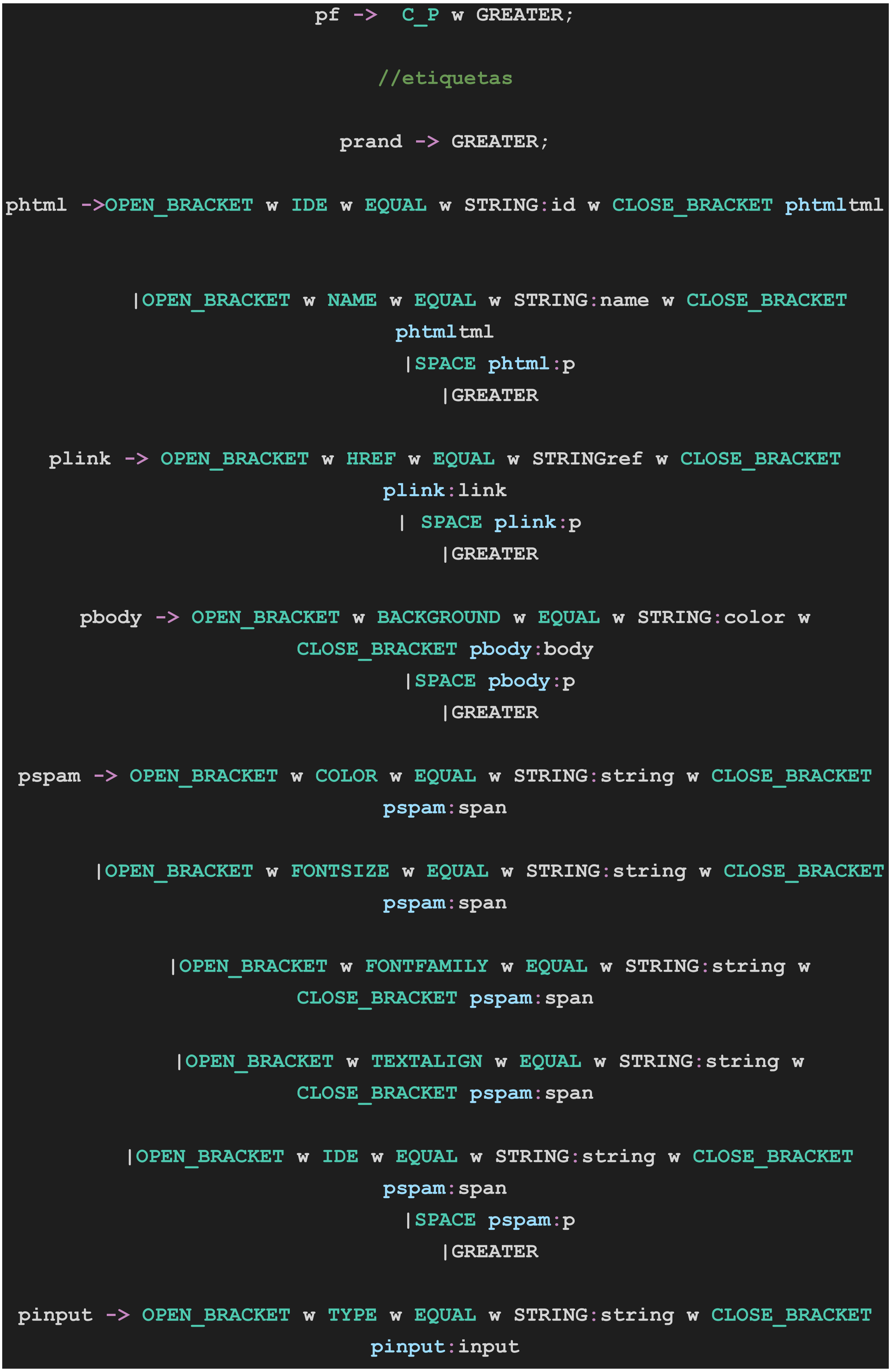
```
                   |plaintetiquete1
                   |plainparemeter1
                      |ONCLICK1
                       |SPACE1
             plaintexto -> texto1

               texto -> GREATER
                       |OPTION
                  |OPEN_PARENTHESIS
                 |CLOSE_PARENTHESIS1
                    |OPEN_BRACKET
                    |CLOSE_BRACKET
                        |EQUAL
                   |plaintetiquete
                   |plainparemeter
                       |ONCLICK
                       |SPACE;

                 //structures

              s -> plaintext1 s2
                 |starttag1   s2
                   |/*empty*/;

              t -> plaintext1 t2
                      |tags

      /*Usar para la cadena del script*/
             o -> plaintexto1 o2
                   |LESS1 o_a2
                     |DIV1 o2

                 o_a ->  DIV
                   |plaintexto1 o2
    startetiquete -> htmls:s t:tags htmlf
                         |etiquete:s

        //left tag will be the object
        etiquete -> heads:tag t1 headf
|titles:tag t1 titlef   Tags((ArrayList<Tag>)r); RESULT =
                   title;:}
                   |links:tag t1 linkf
                   |bodys:tag t1 bodyf
```



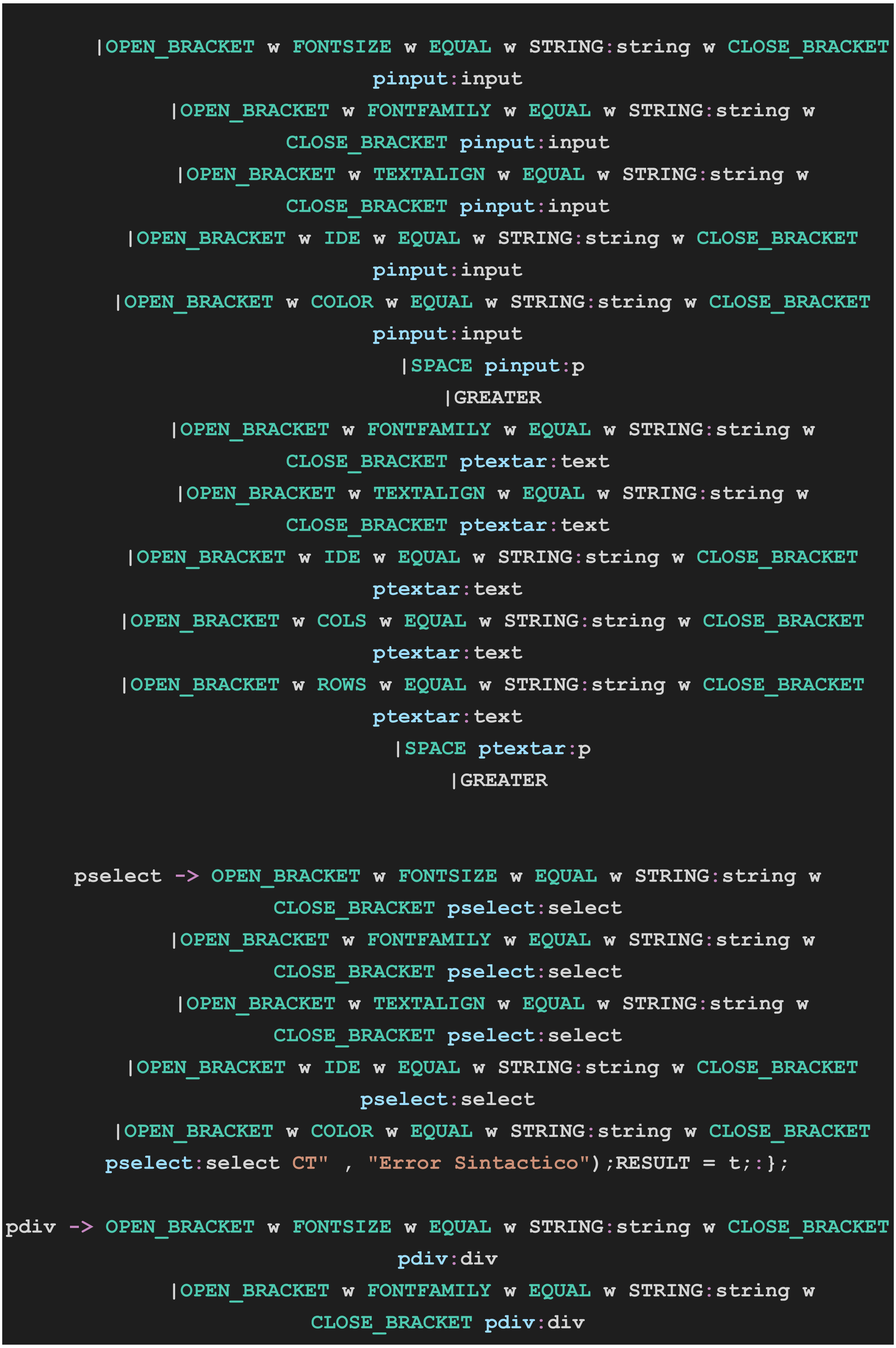
```
                      |spams:tag t1 spamf
                     |inputs:tag t1 inputf
                    |textars:tag t1 textarf
               |selects:tag optionselec1 selectf
                    |scripts:tag o1 scriptf
                       |divs:tag t1 divf
                     |images:tag t1 imagef
                            |brs:tag

         optionselec -> LESS optionselec_re
                      |plaintext1 optionselec2


optionselec_re -> options:op o1 optionf optionselec2
        | options:op o1 error optionselec2 {:RESULT = e2;:}

                                |DIV



                       //Tags

           scripts -> C_SCRIPTING w prand

        scriptf ->  C_SCRIPTING w GREATER ;

               htmls -> C_GCIC phtmlt

            htmlf ->  C_GCIC w GREATER ;

              heads -> C_HEAD w prand

            headf ->  C_HEAD w GREATER ;

             titles -> C_TITLE w prand

           titlef ->  C_TITLE w GREATER;

               links -> C_LINK plinkt

            linkf ->  C_LINK w GREATER;

               bodys -> C_BODY pbodyt
```



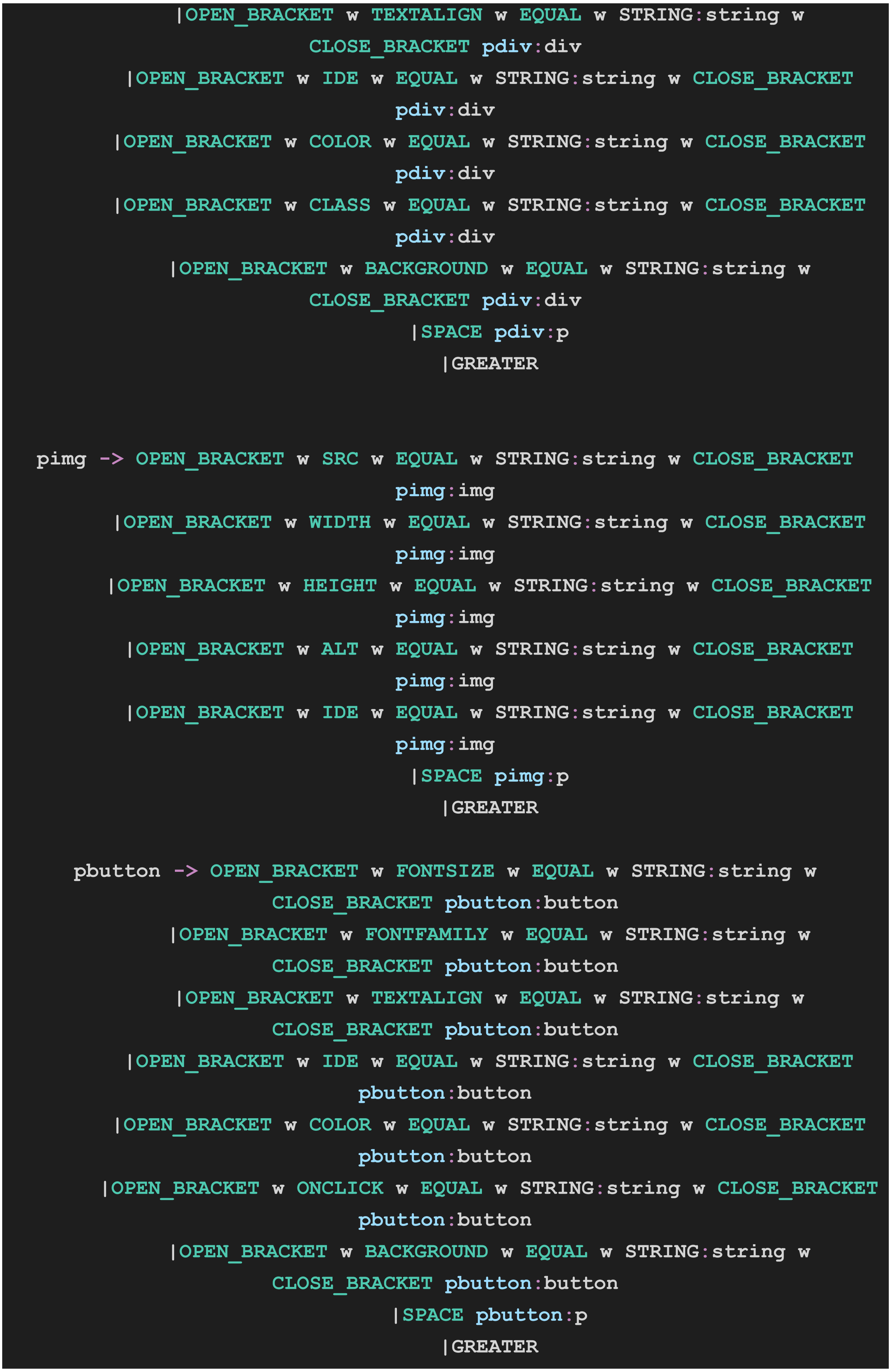
```
bodyf ->  C_BODY w GREATER;

spams -> C_SPAM pspamt

spamf ->  C_SPAM w GREATER;

inputs -> C_INPUT pinputt

inputf ->  C_INPUT w GREATER;

textars -> C_TEXTAREA ptextart

textarf ->  C_TEXTAREA w GREATER

selects -> C_SELECT pselectt

selectf ->  C_SELECT w GREATER;

options -> C_OPTION w prand

optionf ->  C_OPTION w GREATER;

divs -> C_DIV pdivt

divf ->  C_DIV w GREATER;

images -> C_IMG pimgt

imagef ->  C_IMG w GREATER

brs -> C_BR w prand

buttons -> C_BUTTON pbuttont

buttonf ->  C_BUTTON w GREATER;

htexts -> C_H1 phtextt

htextf ->  C_H1 w GREATER;

ps -> C_P ppt
```



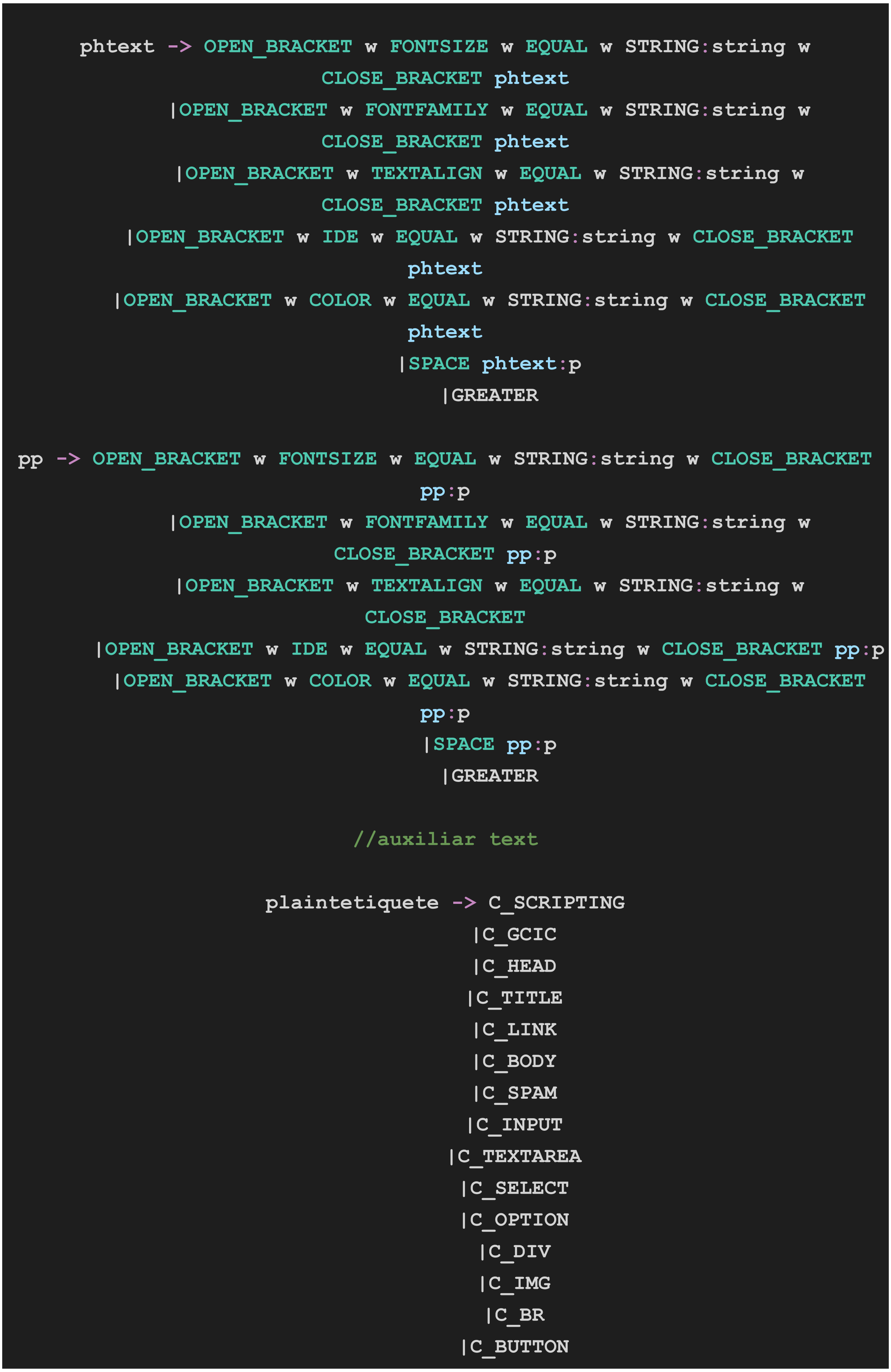
```
pf ->  C_P w GREATER;

//etiquetas

prand -> GREATER;

phtml ->OPEN_BRACKET w IDE w EQUAL w STRING:id w CLOSE_BRACKET phtmltml


|OPEN_BRACKET w NAME w EQUAL w STRING:name w CLOSE_BRACKET
phtmltml
|SPACE phtml:p
|GREATER

plink -> OPEN_BRACKET w HREF w EQUAL w STRINGref w CLOSE_BRACKET
plink:link
| SPACE plink:p
|GREATER

pbody -> OPEN_BRACKET w BACKGROUND w EQUAL w STRING:color w
CLOSE_BRACKET pbody:body
|SPACE pbody:p
|GREATER

pspam -> OPEN_BRACKET w COLOR w EQUAL w STRING:string w CLOSE_BRACKET
pspam:span

|OPEN_BRACKET w FONTSIZE w EQUAL w STRING:string w CLOSE_BRACKET
pspam:span

|OPEN_BRACKET w FONTFAMILY w EQUAL w STRING:string w
CLOSE_BRACKET pspam:span

|OPEN_BRACKET w TEXTALIGN w EQUAL w STRING:string w
CLOSE_BRACKET pspam:span

|OPEN_BRACKET w IDE w EQUAL w STRING:string w CLOSE_BRACKET
pspam:span
|SPACE pspam:p
|GREATER

pinput -> OPEN_BRACKET w TYPE w EQUAL w STRING:string w CLOSE_BRACKET
pinput:input
```



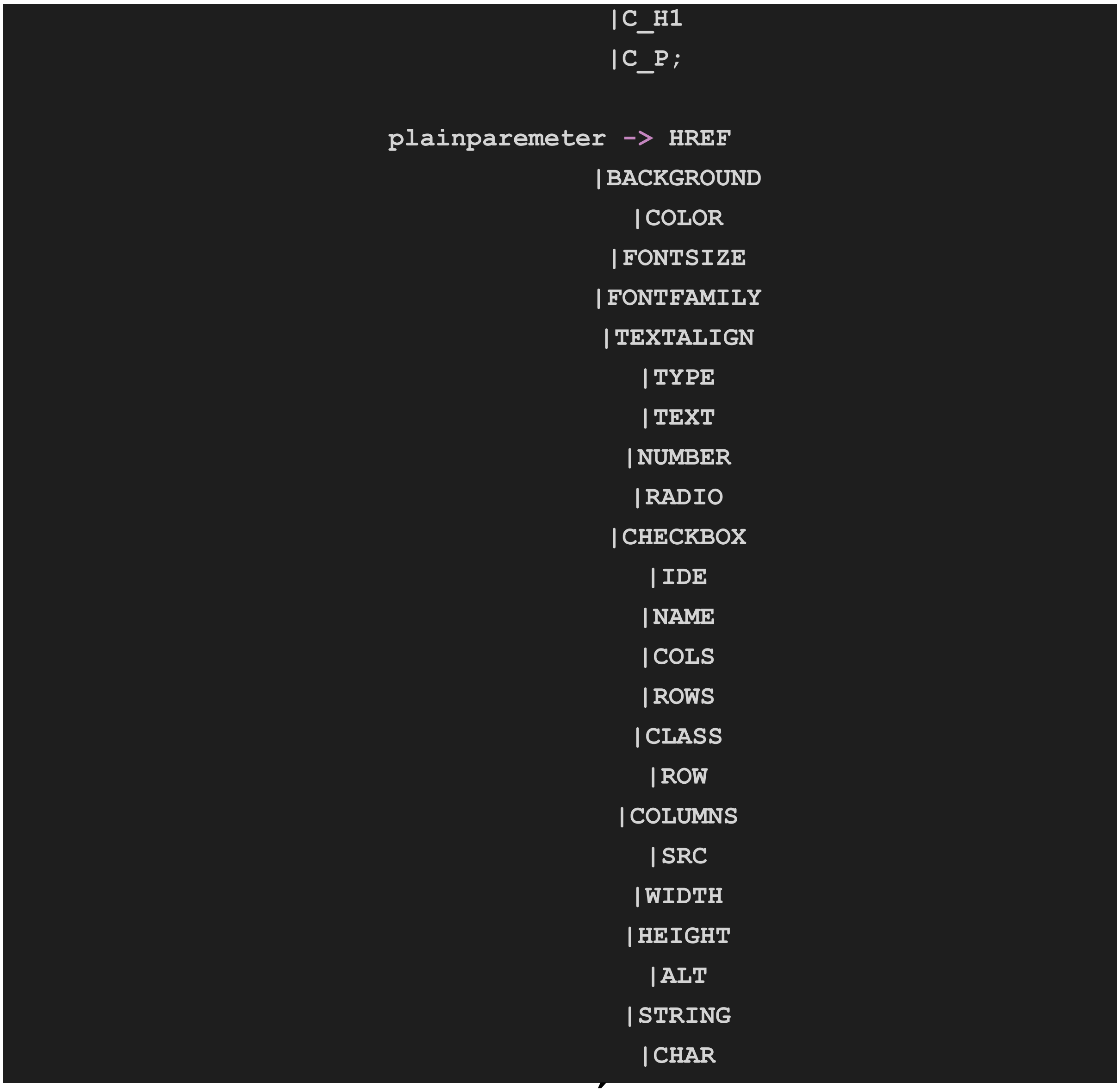
```
       |OPEN_BRACKET w FONTSIZE w EQUAL w STRING:string w CLOSE_BRACKET
                            pinput:input
             |OPEN_BRACKET w FONTFAMILY w EQUAL w STRING:string w
                     CLOSE_BRACKET pinput:input
             |OPEN_BRACKET w TEXTALIGN w EQUAL w STRING:string w
                     CLOSE_BRACKET pinput:input
          |OPEN_BRACKET w IDE w EQUAL w STRING:string w CLOSE_BRACKET
                            pinput:input
         |OPEN_BRACKET w COLOR w EQUAL w STRING:string w CLOSE_BRACKET
                            pinput:input
                              |SPACE pinput:p
                                 |GREATER
             |OPEN_BRACKET w FONTFAMILY w EQUAL w STRING:string w
                     CLOSE_BRACKET ptextar:text
             |OPEN_BRACKET w TEXTALIGN w EQUAL w STRING:string w
                     CLOSE_BRACKET ptextar:text
          |OPEN_BRACKET w IDE w EQUAL w STRING:string w CLOSE_BRACKET
                            ptextar:text
         |OPEN_BRACKET w COLS w EQUAL w STRING:string w CLOSE_BRACKET
                            ptextar:text
         |OPEN_BRACKET w ROWS w EQUAL w STRING:string w CLOSE_BRACKET
                            ptextar:text
                              |SPACE ptextar:p
                                 |GREATER


     pselect -> OPEN_BRACKET w FONTSIZE w EQUAL w STRING:string w
                    CLOSE_BRACKET pselect:select
             |OPEN_BRACKET w FONTFAMILY w EQUAL w STRING:string w
                    CLOSE_BRACKET pselect:select
             |OPEN_BRACKET w TEXTALIGN w EQUAL w STRING:string w
                    CLOSE_BRACKET pselect:select
          |OPEN_BRACKET w IDE w EQUAL w STRING:string w CLOSE_BRACKET
                           pselect:select
         |OPEN_BRACKET w COLOR w EQUAL w STRING:string w CLOSE_BRACKET
        pselect:select CT" , "Error Sintactico");RESULT = t;:};

pdiv -> OPEN_BRACKET w FONTSIZE w EQUAL w STRING:string w CLOSE_BRACKET
                              pdiv:div
             |OPEN_BRACKET w FONTFAMILY w EQUAL w STRING:string w
                       CLOSE_BRACKET pdiv:div
```



```
             |OPEN_BRACKET w TEXTALIGN w EQUAL w STRING:string w
                        CLOSE_BRACKET pdiv:div
         |OPEN_BRACKET w IDE w EQUAL w STRING:string w CLOSE_BRACKET
                               pdiv:div
        |OPEN_BRACKET w COLOR w EQUAL w STRING:string w CLOSE_BRACKET
                               pdiv:div
        |OPEN_BRACKET w CLASS w EQUAL w STRING:string w CLOSE_BRACKET
                               pdiv:div
            |OPEN_BRACKET w BACKGROUND w EQUAL w STRING:string w
                        CLOSE_BRACKET pdiv:div
                                |SPACE pdiv:p
                                  |GREATER


pimg -> OPEN_BRACKET w SRC w EQUAL w STRING:string w CLOSE_BRACKET
                               pimg:img
        |OPEN_BRACKET w WIDTH w EQUAL w STRING:string w CLOSE_BRACKET
                               pimg:img
       |OPEN_BRACKET w HEIGHT w EQUAL w STRING:string w CLOSE_BRACKET
                               pimg:img
         |OPEN_BRACKET w ALT w EQUAL w STRING:string w CLOSE_BRACKET
                               pimg:img
         |OPEN_BRACKET w IDE w EQUAL w STRING:string w CLOSE_BRACKET
                               pimg:img
                                |SPACE pimg:p
                                  |GREATER

  pbutton -> OPEN_BRACKET w FONTSIZE w EQUAL w STRING:string w
                      CLOSE_BRACKET pbutton:button
            |OPEN_BRACKET w FONTFAMILY w EQUAL w STRING:string w
                      CLOSE_BRACKET pbutton:button
             |OPEN_BRACKET w TEXTALIGN w EQUAL w STRING:string w
                      CLOSE_BRACKET pbutton:button
         |OPEN_BRACKET w IDE w EQUAL w STRING:string w CLOSE_BRACKET
                             pbutton:button
        |OPEN_BRACKET w COLOR w EQUAL w STRING:string w CLOSE_BRACKET
                             pbutton:button
       |OPEN_BRACKET w ONCLICK w EQUAL w STRING:string w CLOSE_BRACKET
                             pbutton:button
            |OPEN_BRACKET w BACKGROUND w EQUAL w STRING:string w
                      CLOSE_BRACKET pbutton:button
                               |SPACE pbutton:p
                                  |GREATER
```

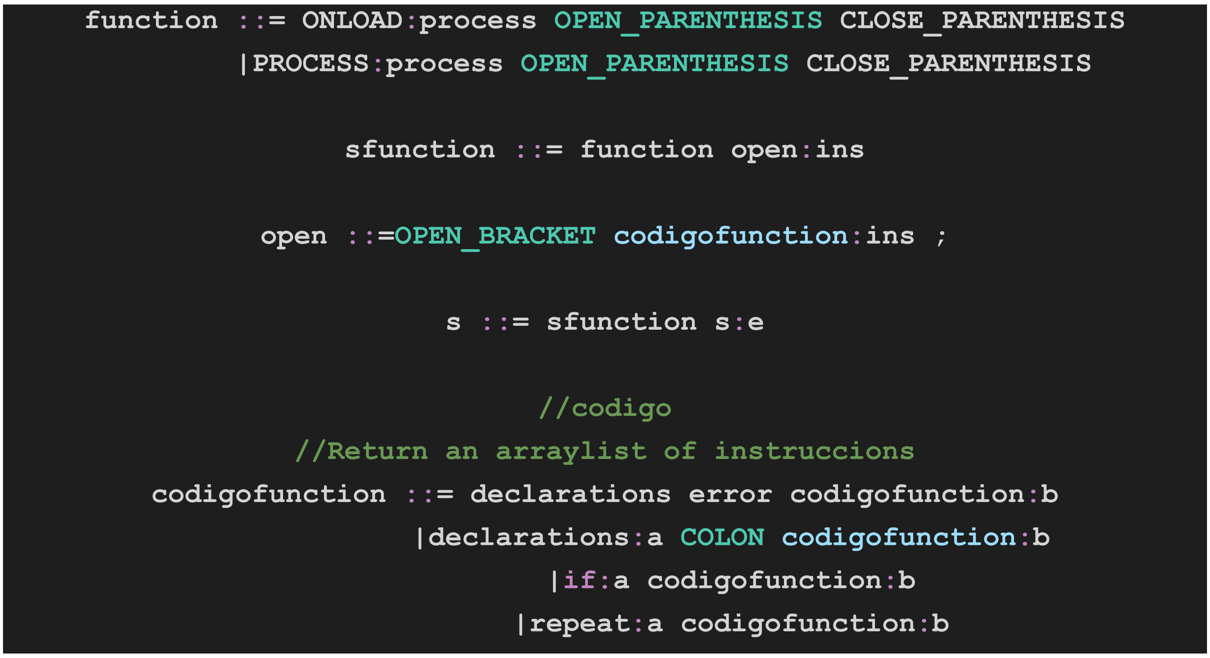


```
    phtext -> OPEN_BRACKET w FONTSIZE w EQUAL w STRING:string w
                        CLOSE_BRACKET phtext
           |OPEN_BRACKET w FONTFAMILY w EQUAL w STRING:string w
                        CLOSE_BRACKET phtext
           |OPEN_BRACKET w TEXTALIGN w EQUAL w STRING:string w
                        CLOSE_BRACKET phtext
        |OPEN_BRACKET w IDE w EQUAL w STRING:string w CLOSE_BRACKET
                              phtext
       |OPEN_BRACKET w COLOR w EQUAL w STRING:string w CLOSE_BRACKET
                              phtext
                             |SPACE phtext:p
                                |GREATER

pp -> OPEN_BRACKET w FONTSIZE w EQUAL w STRING:string w CLOSE_BRACKET
                               pp:p
           |OPEN_BRACKET w FONTFAMILY w EQUAL w STRING:string w
                         CLOSE_BRACKET pp:p
           |OPEN_BRACKET w TEXTALIGN w EQUAL w STRING:string w
                           CLOSE_BRACKET
      |OPEN_BRACKET w IDE w EQUAL w STRING:string w CLOSE_BRACKET pp:p
       |OPEN_BRACKET w COLOR w EQUAL w STRING:string w CLOSE_BRACKET
                               pp:p
                              |SPACE pp:p
                                |GREATER

                           //auxiliar text

                  plaintetiquete -> C_SCRIPTING
                                  |C_GCIC
                                  |C_HEAD
                                 |C_TITLE
                                  |C_LINK
                                  |C_BODY
                                  |C_SPAM
                                 |C_INPUT
                                |C_TEXTAREA
                                 |C_SELECT
                                 |C_OPTION
                                   |C_DIV
                                   |C_IMG
                                   |C_BR
                                 |C_BUTTON
```

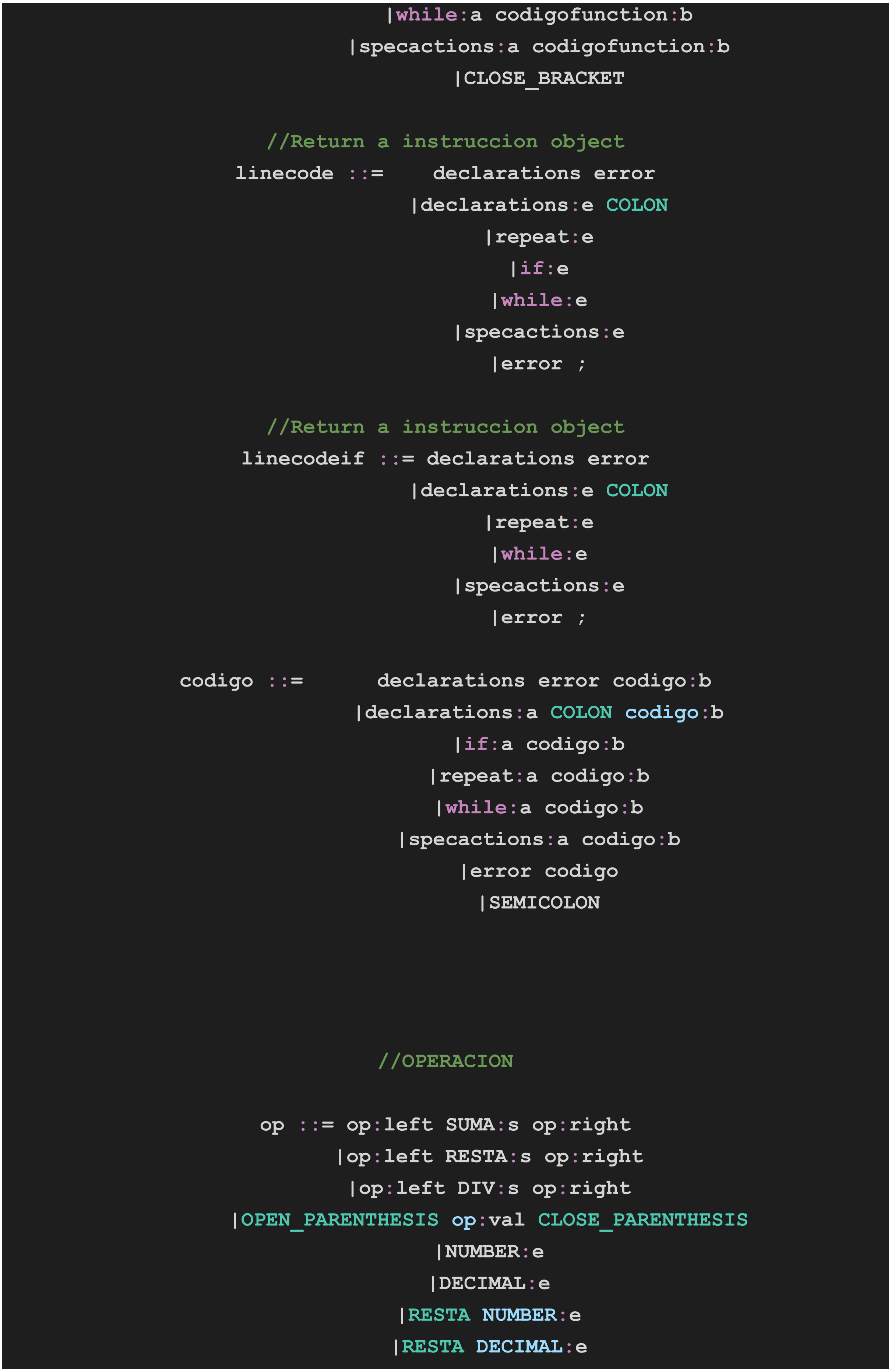


```
              |C_H1
              |C_P;

plainparemeter -> HREF
             |BACKGROUND
                |COLOR
              |FONTSIZE
             |FONTFAMILY
              |TEXTALIGN
                |TYPE
                |TEXT
               |NUMBER
                |RADIO
              |CHECKBOX
                 |IDE
                |NAME
                |COLS
                |ROWS
                |CLASS
                 |ROW
               |COLUMNS
                 |SRC
                |WIDTH
               |HEIGHT
                 |ALT
               |STRING
                |CHAR
```

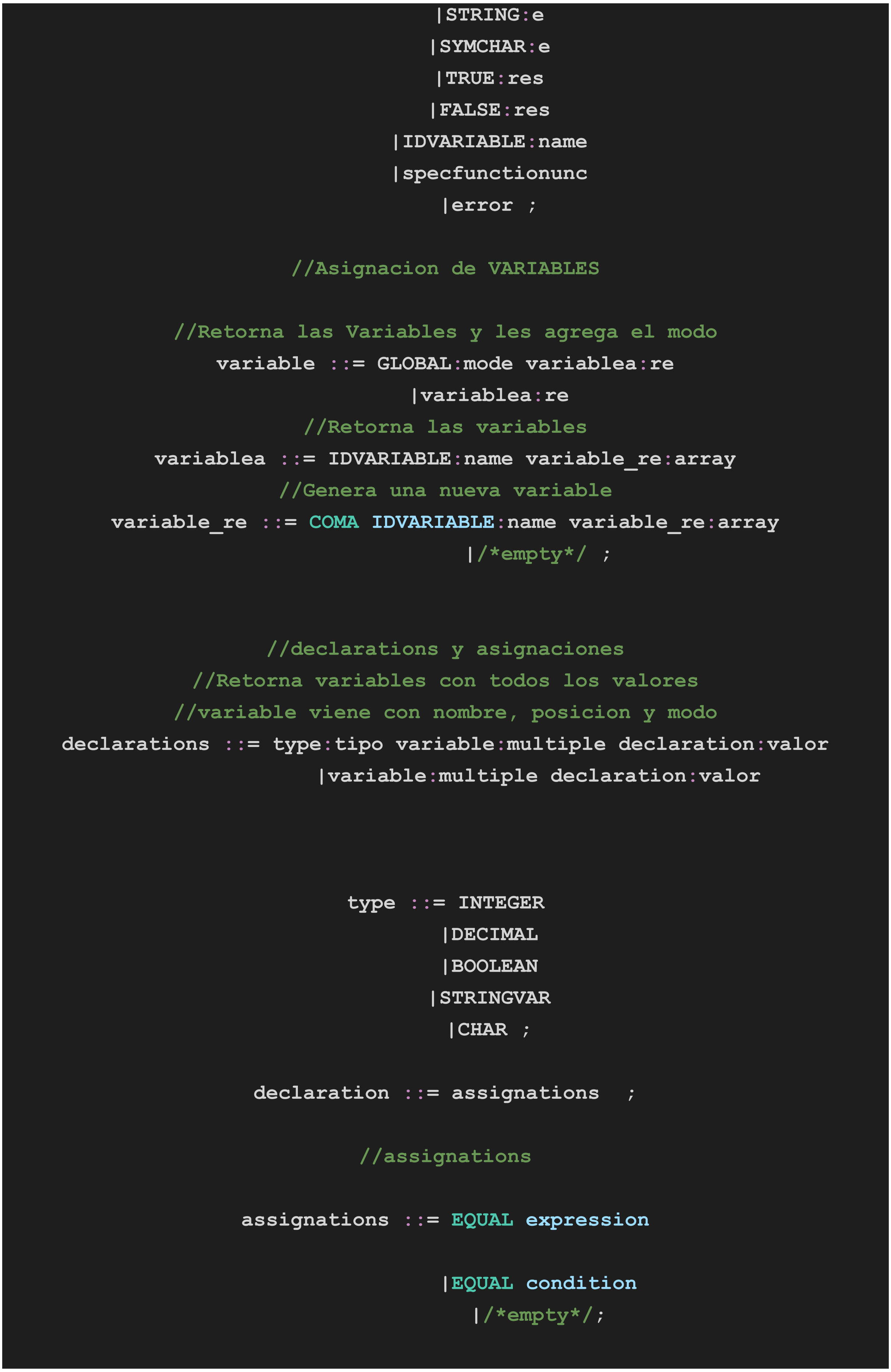
ANALIZADOR SINTÁCTICO SCRIPT



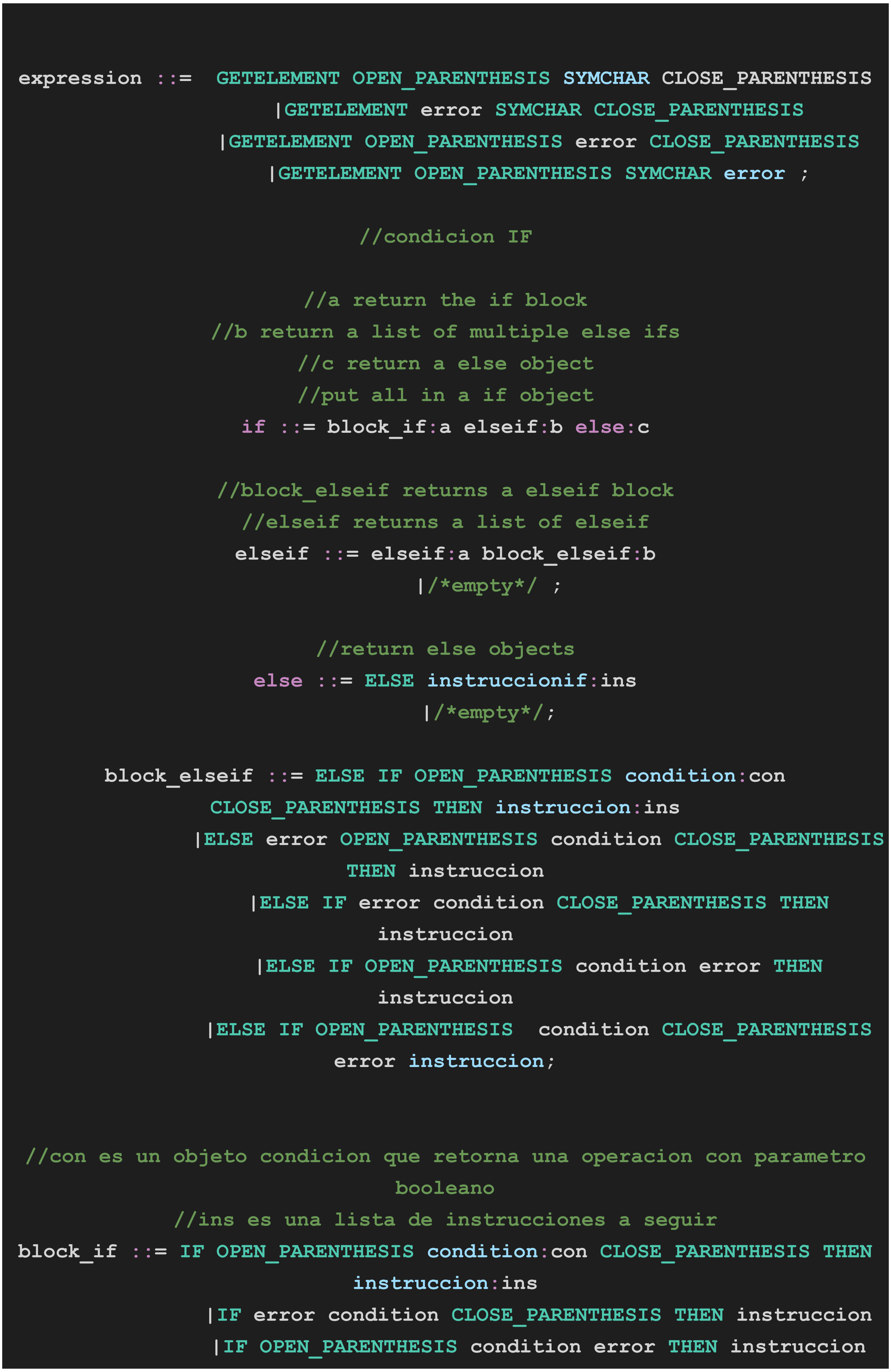
```
function ::= ONLOAD:process OPEN_PARENTHESIS CLOSE_PARENTHESIS
         |PROCESS:process OPEN_PARENTHESIS CLOSE_PARENTHESIS

               sfunction ::= function open:ins

          open ::=OPEN_BRACKET codigofunction:ins ;

                     s ::= sfunction s:e

                           //codigo
            //Return an arraylist of instruccions
    codigofunction ::= declarations error codigofunction:b
                   |declarations:a COLON codigofunction:b
                           |if:a codigofunction:b
                         |repeat:a codigofunction:b
```



```
                               |while:a codigofunction:b
                            |specactions:a codigofunction:b
                                     |CLOSE_BRACKET

                      //Return a instruccion object
                   linecode ::=    declarations error
                                 |declarations:e COLON
                                       |repeat:e
                                         |if:e
                                        |while:e
                                     |specactions:e
                                        |error ;

                      //Return a instruccion object
                    linecodeif ::= declarations error
                                 |declarations:e COLON
                                       |repeat:e
                                        |while:e
                                     |specactions:e
                                        |error ;

              codigo ::=      declarations error codigo:b
                             |declarations:a COLON codigo:b
                                     |if:a codigo:b
                                   |repeat:a codigo:b
                                   |while:a codigo:b
                                |specactions:a codigo:b
                                     |error codigo
                                       |SEMICOLON




                               //OPERACION

                     op ::= op:left SUMA:s op:right
                           |op:left RESTA:s op:right
                            |op:left DIV:s op:right
                   |OPEN_PARENTHESIS op:val CLOSE_PARENTHESIS
                                   |NUMBER:e
                                   |DECIMAL:e
                                |RESTA NUMBER:e
                                |RESTA DECIMAL:e
```

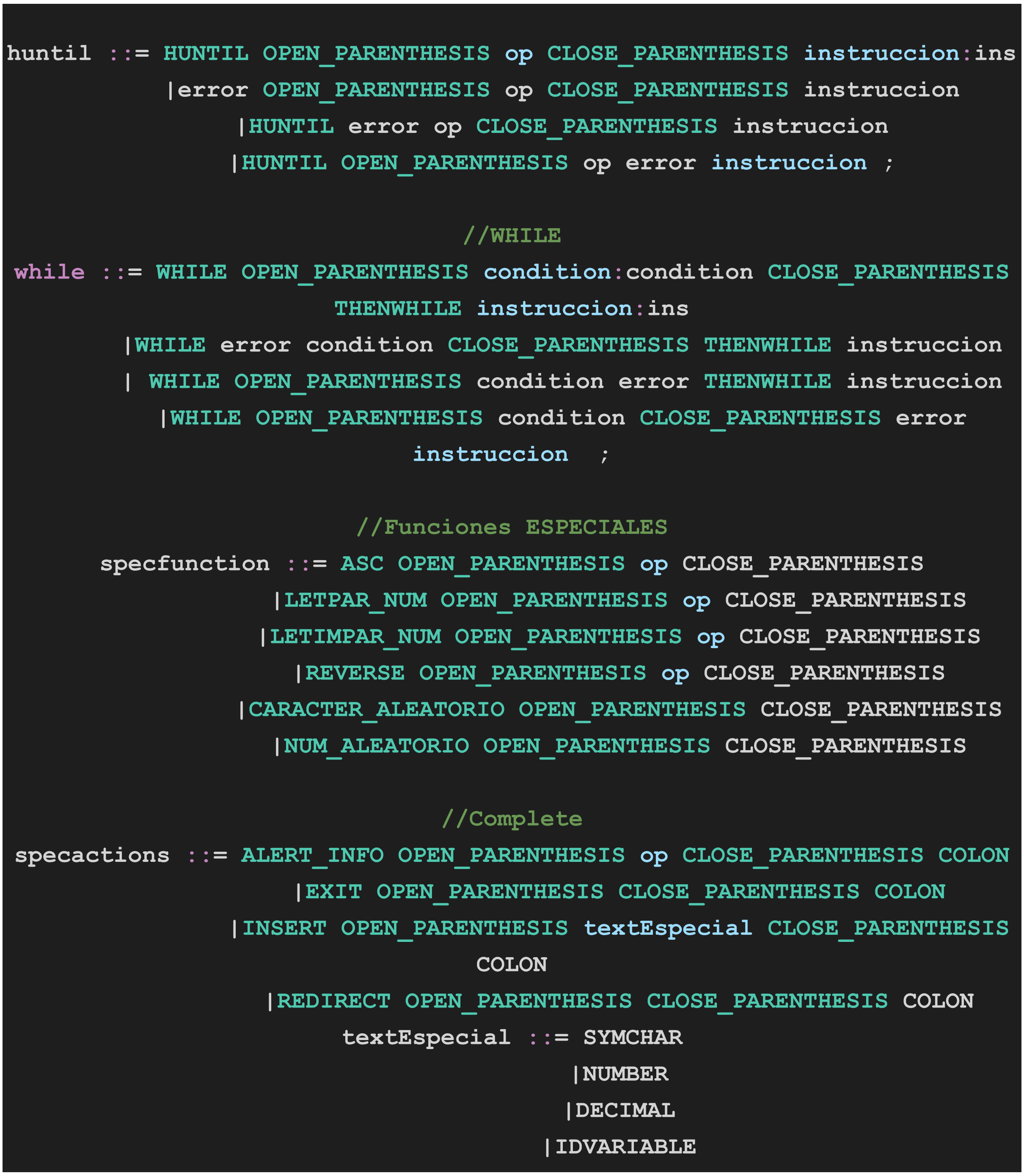


```
                              |STRING:e
                              |SYMCHAR:e
                              |TRUE:res
                              |FALSE:res
                           |IDVARIABLE:name
                           |specfunctionunc
                               |error ;

                   //Asignacion de VARIABLES

         //Retorna las Variables y les agrega el modo
            variable ::= GLOBAL:mode variablea:re
                            |variablea:re
                    //Retorna las variables
       variablea ::= IDVARIABLE:name variable_re:array
                  //Genera una nueva variable
    variable_re ::= COMA IDVARIABLE:name variable_re:array
                                 |/*empty*/ ;


                 //declarations y asignaciones
          //Retorna variables con todos los valores
         //variable viene con nombre, posicion y modo
declarations ::= type:tipo variable:multiple declaration:valor
                     |variable:multiple declaration:valor



                       type ::= INTEGER
                               |DECIMAL
                               |BOOLEAN
                              |STRINGVAR
                               |CHAR ;

                declaration ::= assignations  ;

                        //assignations

               assignations ::= EQUAL expression

                               |EQUAL condition
                                 |/*empty*/;
```



```
expression ::=  GETELEMENT OPEN_PARENTHESIS SYMCHAR CLOSE_PARENTHESIS
                     |GETELEMENT error SYMCHAR CLOSE_PARENTHESIS
                |GETELEMENT OPEN_PARENTHESIS error CLOSE_PARENTHESIS
                    |GETELEMENT OPEN_PARENTHESIS SYMCHAR error ;

                            //condicion IF

                       //a return the if block
                //b return a list of multiple else ifs
                      //c return a else object
                      //put all in a if object
                  if ::= block_if:a elseif:b else:c

                //block_elseif returns a elseif block
                  //elseif returns a list of elseif
                 elseif ::= elseif:a block_elseif:b
                                |/*empty*/ ;

                        //return else objects
                   else ::= ELSE instruccionif:ins
                                 |/*empty*/;

       block_elseif ::= ELSE IF OPEN_PARENTHESIS condition:con
               CLOSE_PARENTHESIS THEN instruccion:ins
              |ELSE error OPEN_PARENTHESIS condition CLOSE_PARENTHESIS
                          THEN instruccion
                   |ELSE IF error condition CLOSE_PARENTHESIS THEN
                             instruccion
                   |ELSE IF OPEN_PARENTHESIS condition error THEN
                             instruccion
               |ELSE IF OPEN_PARENTHESIS  condition CLOSE_PARENTHESIS
                         error instruccion;


 //con es un objeto condicion que retorna una operacion con parametro
                              booleano
            //ins es una lista de instrucciones a seguir
block_if ::= IF OPEN_PARENTHESIS condition:con CLOSE_PARENTHESIS THEN
                           instruccion:ins
               |IF error condition CLOSE_PARENTHESIS THEN instruccion
                |IF OPEN_PARENTHESIS condition error THEN instruccion
```

```
                 |IF OPEN_PARENTHESIS condition CLOSE_PARENTHESIS error
                              instruccion ;


                              //condicion

                comparation ::= op:a EQUALIZATION:s op:b
                             |op:a DIFFERENTIATION:s op:b
                                |op:a LESSTHAN:s op:b
                                  |op:a LESS:s op:b
                               |op:a GREATERTHAN:s op:b
                                 |op:a GREATER:s op:b
                                       |op:e ;

               condition ::= condition:a AND:s condition:b
                            |condition:a OR:s condition:b
                                 |NOT:s condition:a
                    |OPEN_PARENTHESIS condition:e CLOSE_PARENTHESIS
                                  |comparation:a ;


                             //instruccion
   instruccion ::= INIT OPEN_CURLY SEMICOLON codigo:a  CLOSE_CURLY END
                     |INIT error SEMICOLON codigo  CLOSE_CURLY END
                    |INIT OPEN_CURLY error codigo  CLOSE_CURLY END
                   |INIT OPEN_CURLY SEMICOLON error  CLOSE_CURLY END
                     |INIT OPEN_CURLY SEMICOLON codigo  error END
                  |INIT OPEN_CURLY SEMICOLON codigo  CLOSE_CURLY error
                                    |linecode:a

  instruccionif ::= INIT OPEN_CURLY SEMICOLON codigo:a CLOSE_CURLY END
                     |INIT error SEMICOLON codigo  CLOSE_CURLY END
                    |INIT OPEN_CURLY error codigo  CLOSE_CURLY END
                   |INIT OPEN_CURLY SEMICOLON error  CLOSE_CURLY END
                     |INIT OPEN_CURLY SEMICOLON codigo  error END
                  |INIT OPEN_CURLY SEMICOLON codigo  CLOSE_CURLY error
                                   |linecodeif:a

                               //REPEAT
    repeat ::= REPEAT OPEN_PARENTHESIS declarations CLOSE_PARENTHESIS
                               huntil:hun
              |REPEAT error declarations CLOSE_PARENTHESIS huntil
             |REPEAT OPEN_PARENTHESIS declarations error huntil ;
```



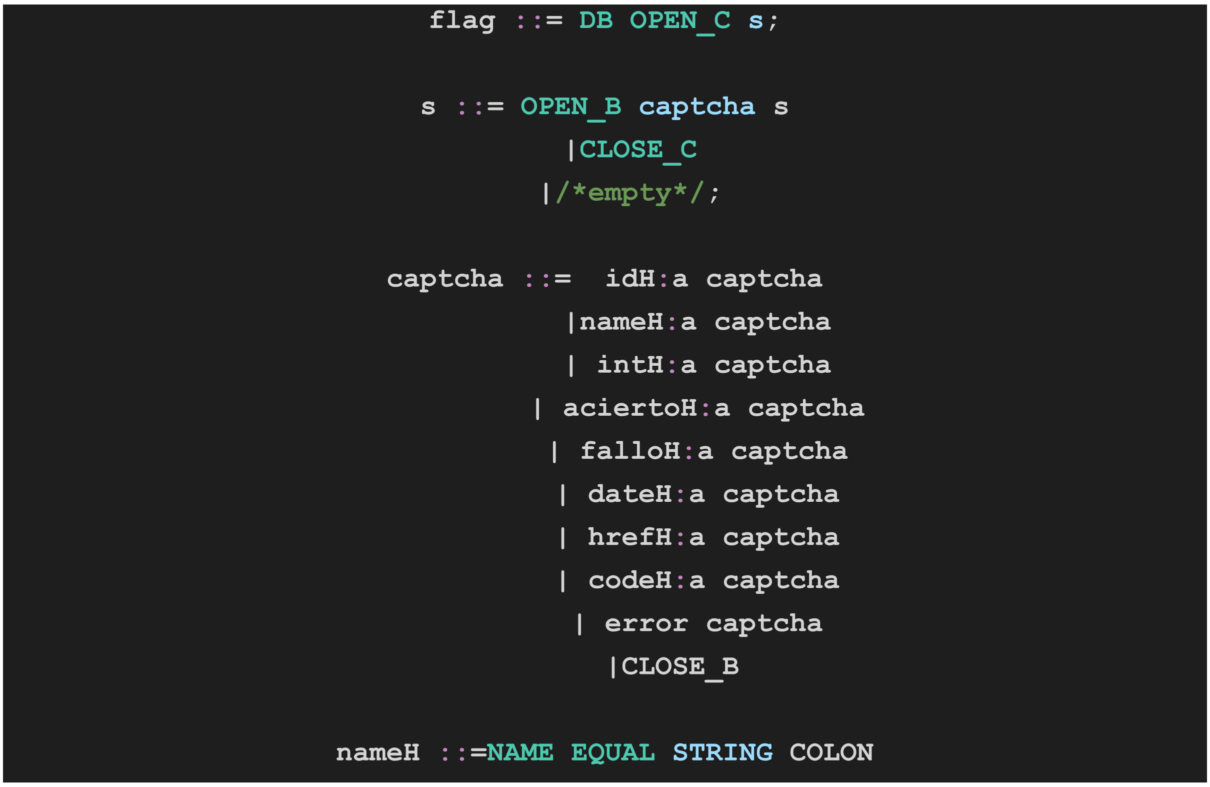
```
huntil ::= HUNTIL OPEN_PARENTHESIS op CLOSE_PARENTHESIS instruccion:ins
                |error OPEN_PARENTHESIS op CLOSE_PARENTHESIS instruccion
                    |HUNTIL error op CLOSE_PARENTHESIS instruccion
                    |HUNTIL OPEN_PARENTHESIS op error instruccion ;

                              //WHILE
while ::= WHILE OPEN_PARENTHESIS condition:condition CLOSE_PARENTHESIS
                    THENWHILE instruccion:ins
          |WHILE error condition CLOSE_PARENTHESIS THENWHILE instruccion
          | WHILE OPEN_PARENTHESIS condition error THENWHILE instruccion
            |WHILE OPEN_PARENTHESIS condition CLOSE_PARENTHESIS error
                              instruccion  ;

                        //Funciones ESPECIALES
     specfunction ::= ASC OPEN_PARENTHESIS op CLOSE_PARENTHESIS
                     |LETPAR_NUM OPEN_PARENTHESIS op CLOSE_PARENTHESIS
                    |LETIMPAR_NUM OPEN_PARENTHESIS op CLOSE_PARENTHESIS
                      |REVERSE OPEN_PARENTHESIS op CLOSE_PARENTHESIS
                  |CARACTER_ALEATORIO OPEN_PARENTHESIS CLOSE_PARENTHESIS
                     |NUM_ALEATORIO OPEN_PARENTHESIS CLOSE_PARENTHESIS

                              //Complete
specactions ::= ALERT_INFO OPEN_PARENTHESIS op CLOSE_PARENTHESIS COLON
                      |EXIT OPEN_PARENTHESIS CLOSE_PARENTHESIS COLON
                  |INSERT OPEN_PARENTHESIS textEspecial CLOSE_PARENTHESIS
                               COLON
                    |REDIRECT OPEN_PARENTHESIS CLOSE_PARENTHESIS COLON
                        textEspecial ::= SYMCHAR
                                     |NUMBER
                                     |DECIMAL
                                     |IDVARIABLE
```

ANALIZADOR LEXICO SAVE

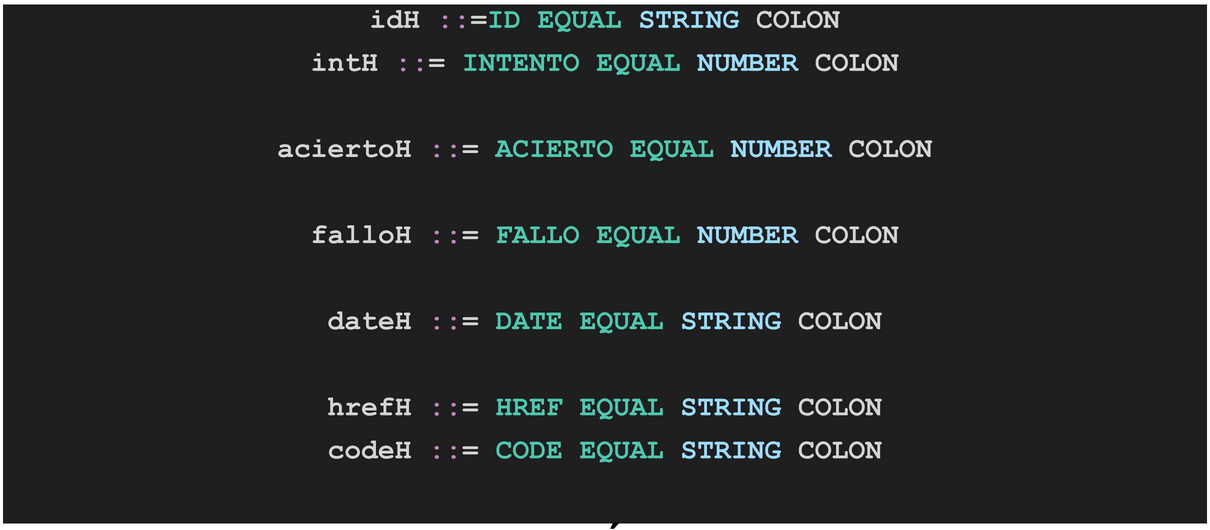
TOKEN	LEXEMA
DB	db.captcha
EQUAL	=
OPEN_C	{

CLOSE_C	}
OPEN_B	[
CLOSE_B	]
COLON	;
ID	ID
INTENTO	INTENTOS
NAME	NAME
ACIERTO	ACIERTOS
FALLO	FALLOS
DATE	DATE
HREF	HREF
CODE	CODE
NÚMERO	[0-9]+
STRING	\"[^\"]*\"

ANALIZADOR SINTÁCTICO SAVE



```
flag ::= DB OPEN_C s;

s ::= OPEN_B captcha s
      |CLOSE_C
      |/*empty*/;

captcha ::=  idH:a captcha
             |nameH:a captcha
             | intH:a captcha
             | aciertoH:a captcha
             | falloH:a captcha
             | dateH:a captcha
             | hrefH:a captcha
             | codeH:a captcha
             | error captcha
             |CLOSE_B

nameH ::=NAME EQUAL STRING COLON
```



```
idH ::=ID EQUAL STRING COLON
intH ::= INTENTO EQUAL NUMBER COLON

aciertoH ::= ACIERTO EQUAL NUMBER COLON

falloH ::= FALLO EQUAL NUMBER COLON

dateH ::= DATE EQUAL STRING COLON

hrefH ::= HREF EQUAL STRING COLON
codeH ::= CODE EQUAL STRING COLON
```

DATOS TÉCNICOS

Se utilizó ECMAScript 6 y JDK16 con servidor TOMCAT 9