# Manual Técnico

# Analizador Sintáctico

Entorno de desarrollo: IDE NETBEANS 11.3
Sistema operativo: Windows 10 X64
Version SO: 20H2
Memoria Ram: 12GB
CPU: intel core !7-7700HQ

# Tabla de Contenidos

# Gramática del lenguaje GCIC

## Gramática de analizador léxico

Números decimales:
decimal=([0]|[1-9][0-9]*)[\.][0-9][0-9]?[0-9]?[0-9]?
Números enteros:
entero = [0-9]+
Carácter encerrado entre comillas simples:
char = [\'][^\'][\']
Una cadena de texto encerrado en comillas simples:
id_element = [\'](^\'])*[\']
Una cadena encerrado entre comillas dobles:
cadena = [\"](^\"])*[\"]
Cadena que no contiene ningún tipo de espacio:
cadena_sin_espacios = [\"](^\"\r\n\t\f])*[\"]
Sucesión de caracteres:
string_total = [a-zA-Z\?\¿\.\@_]+
Sucesión de caracteres que inicia con una letra:
id_variable = [a-zA-Z][a-zA-Z0-9_]*
Comentario de linea:

comentario_linea = "!!"[^\r\n\t\f]+
Comentario de bloque:
comentario_bloque = "<!--"([^\-]|"-"[^\-] |"--"[^\>])*"-->"

Palabras reservadas que no importan las mayúsculas y minúsculas:
C_GCIC = [cC][_][gG][cC][iI][cC]
C_HEAD = [cC][_][hH][eE][aA][dD]
C_TITLE =  [cC][_][tT][iI][tT][lL][eE]
C_LINK =  [cC][_][lL][iI][nN][kK]
C_BODY = [cC][_][bB][oO][dD][yY]
C_SPAM = [cC][_][sS][pP][aA][mM]
C_INPUT =  [cC][_][iI][nN][pP][uU][tT]
C_TEXTAREA = [cC][_][tT][eE][xX][tT][aA][rR][eE][aA]
C_SELECT = [cC][_][sS][eE][lL][eE][cC][Tt]
C_OPTION = [cC][_][oO][pP][tT][iI][oO][nN]
C_DIV = [cC][_][dD][iI][vV]
C_IMG = [cC][_][iI][mM][gG]
C_BR = [cC][_][bB][rR]
C_BUTTON = [cC][_][bB][uU][tT][tT][oO][nN]
C_H1 = [cC][_][hH][1]
C_P = [cC][_][pP]
C_SCRIPTING = [cC][_][sS][cC][rR][iI][pP][tT][iI][nN][gG]

Nombre de una función:
process_ = "PROCESS_"

nombre_funcion = {process_}([a-zA-Z]*)

Palabras reservadas:
href
background
color
Font-size
Font-family
Text-align
type
id
name
cols
rows
class
src
width
height
alt
Onclick
ON_LOAD
INSERT
getElemenById
ASC
DESC

LETPAR_NUM
LETIMPAR_NUM
REVERSE
CARACTER_ALEATORIO
NUM_ALEATORIO
ALERT_INFO
EXIT
@global
integer
string
boolean
decimal
char
true
false
IF
ELSE
THEN
INIT
END
REPEAT
HUNTIL
WHILE
THENWHILE

Símbolos(No cuentan las comillas):

"/"
"{"
"}"
":"
"["
"]"
"("
")"
","
"<"
">"
"!"
"="
"=="
"<="
">="
"!="
";"
"|"
"+"
"-"
"*"
"&"

**Gramática para el analizador Sintactico:**

Símbolos terminales:
MENQ, C_GCIC, CORI, PARAMETRO, IGUAL,STRING, STRING_WS, CORD,MAQ, DIAG,
C_HEAD,C_LINK,C_TITLE,STRING_TOTAL,ID_VARIABLE,C_BODY,C_SPAM,
C_INPUT,C_TEXTAREA,C_SELECT,C_OPTION,C_DIV,C_IMG,C_BR,C_BUTTON,C_H1,C_P,C_SCRIPTING,
ON_LOAD,NOMBRE_FUNCION,PAI,PAD,LLAI, LLAD, DOS_PUNTOS,COMA,ADMIRACION,
SEMI,BARRA90,AND_1,SUMA,MENOS,MULT,INSERT,GET_ELEMENT_BY_ID,IGUALACION,MEN_QUE,
MAY_QUE,DIFERENTE,CHAR,ENTERO,DECIMAL,BOOLEAN,INVALID,ASC,DESC,LETPAR_NUM,
LETIMPAR_NUM,REVERSE,CARACTER_ALEATORIO,NUM_ALEATORIO,ALERT_INFO,EXIT,REDIRECT
GLOBAL,ID_INTEGER,ID_STRING,ID_BOOLEAN,ID_DECIMAL,ID_CHAR, IF,ELSE,THEN,INIT,END,
REPEAT,HUNTIL,WHILE,THENWHILE, ID_ELEMENT

Símbolos no terminales:
inicio,etiqueta_gcic,inicio_gcic,parametros_etiquetas,varios_parametro_etiqueta,parametro_etiqueta,
inicio_param_etiq,string,fin_param_etiq,cierre_etiqueta,fin_etiqueta_gcic,pre_fin_etiq,cierre_gcic,head,etiqueta_head,
inicio_head,fin_etiqueta_head,cierre_head,link,etiqueta_link,inicio_link,fin_etiqueta_link,cierre_link,title,
etiqueta_title,inicio_title,fin_etiqueta_title,cierre_title,cont_etiqueta,varios_contenido,todo_contenido,cont_head,
etiquetas_para_head,etiqueta_para_head,body,etiqueta_body,inicio_body,fin_etiqueta_body,cierre_body,cont_body,
etiquetas_para_body,etiqueta_para_body,c_etiqueta_g,etq_generica,etiqueta_g,cont_etq_g,varios_contenido_g,
un_cont_etq_g,inicio_g, fin_etiqueta_g,cierre_g,etiqueta_br,inicio_br,scripting,etiqueta_scripting,inicio_scripting,
fin_etiqueta_scripting,cierre_scripting,cont_scripting,funciones,functions,on_load,process_,funcion_on_load,
name_on_load,parentesis,funcion_process_,name_funcion_g,cont_funcion,inicio_funcion,fin_funcion,funcion,
tipos_instrucciones,tipo_instruccion,expr_list, expr,estruc_init,inicio_init,el_init,fin_estruct,ini_estruct,
declaracion_asginacion,fin_instruccion,declaracion,tipo_dato,modo,nombre_variable,

otras_variables,otra_variable,nombre_otra_variable,id_variable,asignacion,tipos_datos,cont_element_by_id,
estruc_if,if,condicion,condicion_f,cont_estruc_if,estruc_else_if,else_if,un_else_if,inicio_else_if,estruc_else,else,inicio_else,
fin_instruccion_error,funciones_especiales,funciones_especiales_parametros,funciones_especiales_sin_parametros,
funciones_especiales_sin_tipo_return,struc_repeat,asig_decla_repeat,declaracion_asginacion_repeat,declaracion_repeat,
asignacion_repeat,huntil,estruc_while,thenwhile,insert,estruc_cont_insert,cont_insert,instrucciones,instrucciones_para_if

Precedencia:
precedence left SUMA, MENOS,COMA;
precedence left MULT,DIAG;
precedence left IGUALACION,MEN_QUE,MAY_QUE,DIFERENTE,MAQ,MENQ;
precedence left BARRA90;
precedence left AND_1;
precedence left ADMIRACION;

Gramática:

```
inicio ::= etiqueta_gcic head body fin_etiqueta_gcic
            ;

etiqueta_gcic ::= inicio_gcic parametros_etiquetas
                    ;

inicio_gcic ::= MENQ C_GCIC
                |error
                ;
```

```
parametros_etiquetas ::= varios_parametro_etiqueta MAQ
                        |MAQ
                        |error
                        ;

varios_parametro_etiqueta ::=    varios_parametro_etiqueta parametro_etiqueta
                                |parametro_etiqueta
                                ;

parametro_etiqueta ::= inicio_param_etiq string fin_param_etiq
                        ;

inicio_param_etiq ::= CORI PARAMETRO IGUAL
                        ;

string ::=   STRING
             |STRING_WS
             |error
             ;

fin_param_etiq ::=      CORD
                       |error
                       ;
```

```
cierre_etiqueta ::= MAQ
                    |error
                    ;

fin_etiqueta_gcic ::=  pre_fin_etiq cierre_gcic
                       |error
                       ;
pre_fin_etiq ::= MENQ DIAG
                 ;
cierre_gcic ::= C_GCIC MAQ
                |error
                ;


head ::= etiqueta_head cont_head fin_etiqueta_head
         ;
etiqueta_head ::=  inicio_head cierre_etiqueta
                   ;
inicio_head ::= MENQ C_HEAD
                |error
                ;
fin_etiqueta_head ::=       pre_fin_etiq cierre_head
                       |error C_HEAD
                       ;
cierre_head ::= C_HEAD MAQ
```

```
                    |error
                    ;

cont_head ::=  cont_head etiqueta_para_head
                    |etiqueta_para_head
                    ;

etiquetas_para_head ::= etiquetas_para_head etiqueta_para_head
                         |etiqueta_para_head
                         ;

etiqueta_para_head ::= link
                        |title
                        |error parametros_etiquetas
                        ;

link ::= etiqueta_link fin_etiqueta_link
          ;
etiqueta_link ::=    inicio_link parametros_etiquetas
                        ;
inicio_link ::= MENQ C_LINK
                    ;
fin_etiqueta_link ::= pre_fin_etiq cierre_link
                           |error C_LINK
                           ;
```

```
cierre_link ::= C_LINK MAQ
                |error
                ;

title ::= etiqueta_title cont_etiqueta fin_etiqueta_title
          ;

etiqueta_title ::= inicio_title cierre_etiqueta
                   ;
inicio_title ::= MENQ C_TITLE
                 ;
fin_etiqueta_title ::= pre_fin_etiq cierre_title
                       |error
                       ;
cierre_title ::=    C_TITLE MAQ
                    |error
                    ;

cont_etiqueta ::=       varios_contenido
                    |
                    ;

varios_contenido ::= varios_contenido todo_contenido
                    |todo_contenido
                    ;
```

```
todo_contenido ::= STRING
                  |STRING_WS
                  |STRING_TOTAL
                  |ID_VARIABLE|ID_ELEMENT
                  |MULT
                  |SUMA
                  |MENOS
                  |BARRA90|DIAG|CORD|CORI|IGUAL
                  |LLAI|LLAD|COMA|SEMI|DOS_PUNTOS|ADMIRACION|INSERTAR|AND_1
                  |PAI|PAD|CHAR|ENTERO|DECIMAL|BOOLEAN|INVALID
                  ;

body ::= etiqueta_body cont_body fin_etiqueta_body
        ;

etiqueta_body ::= inicio_body parametros_etiquetas
                  ;

inicio_body ::= MENQ C_BODY
               |error
               ;
fin_etiqueta_body ::=      pre_fin_etiq cierre_body
                          |error C_BODY
                          ;
```

```
cierre_body ::= C_BODY MAQ
                |error
                ;
cont_body ::=       cont_body etiqueta_para_body
                |etiqueta_para_body
                ;

etiquetas_para_body ::= etiquetas_para_body etiqueta_para_body
                    |etiqueta_para_body
                    ;

etiqueta_para_body ::=     etq_generica
                        |etiqueta_br
                        |scripting
                        |error parametros_etiquetas
                        ;
c_etiqueta_g ::=          C_SPAM
                        |C_INPUT
                        |C_TEXTAREA
                        |C_SELECT
                        |C_OPTION
                        |C_DIV
                        |C_IMG
                        |C_BUTTON
                        |C_H1
```

```
                        |C_P
                        ;

etq_generica ::= etiqueta_g cont_etiqueta fin_etiqueta_g
            ;
etiqueta_g ::=      inicio_g parametros_etiquetas
                    ;
inicio_g ::= MENQ c_etiqueta_g
                ;
fin_etiqueta_g ::=      pre_fin_etiq cierre_g
                    |error
                    ;
cierre_g ::=    c_etiqueta_g MAQ
                |error
                ;

etiqueta_br ::= inicio_br cierre_etiqueta
                ;
inicio_br ::= MENQ C_BR
                ;

scripting ::=   etiqueta_scripting cont_scripting
                ;
etiqueta_scripting ::= inicio_scripting cierre_etiqueta
                        ;
```

```
inicio_scripting ::= MENQ C_SCRIPTING
                     ;
fin_etiqueta_scripting ::= pre_fin_etiq cierre_scripting
                           |error C_SCRIPTING
                           ;
cierre_scripting ::=     C_SCRIPTING MAQ
                         |error
                         ;

cont_scripting ::= funciones fin_etiqueta_scripting
                   |fin_etiqueta_scripting
                   ;
funciones ::= funciones functions
              |functions
              ;
functions ::= on_load
          |process_
          |error parentesis
          ;

on_load ::= funcion_on_load cont_funcion
          ;
process_ ::= funcion_process_ cont_funcion
          ;
```

```
funcion_on_load ::= name_on_load parentesis
                    ;

name_on_load ::= ON_LOAD
                 ;
parentesis ::= PAI PAD
               |error
               ;

funcion_process_ ::= name_funcion_g parentesis
                     ;
name_funcion_g ::= NOMBRE_FUNCION
                   ;

cont_funcion ::= inicio_funcion funcion fin_funcion
                 ;

inicio_funcion ::= CORI
                   |error
                   ;
fin_funcion ::= CORD
                |error
                ;
funcion ::= tipos_instrucciones
          |
```

```
            ;
tipos_instrucciones ::= tipos_instrucciones tipo_instruccion
                        |tipo_instruccion
                        ;
tipo_instruccion ::= declaracion_asginacion
                    |estruc_if
                    |funciones_especiales
                    |estruc_repeat
                    |estruc_while
                    |insert
                    ;
expr          ::= expr:e1 SUMA    expr:e2
                  |expr:e1 MENOS expr:e2
                  |expr:e1 MULT expr:e2
                  |expr:e1 DIAG expr:e2
                  |expr:e1 IGUALACION expr:e2
                  |expr:e1 DIFERENTE expr:e2
                  |expr:e1 MEN_QUE expr:e2
                  |expr:e1 MAY_QUE expr:e2
                  |expr:e1 MAQ expr:e2
                  |expr:e1 MENQ expr:e2
                  |expr:e1 BARRA90 BARRA90 expr:e2
                  |expr:e1 AND_1 AND_1 expr:e2
                  |ADMIRACION expr:e1
                  |PAI expr:e PAD
```

```
                    |MENOS ENTERO:n
                    |MENOS DECIMAL:d
                    |ENTERO:n
                    |DECIMAL:d
                    |CHAR:c
                    |STRING:s
                    |STRING_WS:s
                    |BOOLEAN:b
                    |ID_VARIABLE
                    |funciones_especiales_parametros
                    |funciones_especiales_sin_parametros
                    |error
                     ;

estruc_init ::= inicio_init funcion fin_estruct
                    ;
inicio_init ::= el_init ini_estruct
                    ;
el_init ::= INIT
          |error
          ;
ini_estruct ::= LLAI DOS_PUNTOS
                    |error
                    ;
fin_estruct ::= DOS_PUNTOS LLAD END
```

```
                    |error
                    ;

declaracion_asginacion ::= declaracion asignacion
                                ;
fin_instruccion ::= SEMI
                        ;
declaracion ::= tipo_dato modo
                    |nombre_variable
                    ;
tipo_dato ::= ID_INTEGER
                |ID_DECIMAL
                |ID_STRING
                |ID_BOOLEAN
                |ID_CHAR
                ;
modo ::= GLOBAL nombre_variable
        |nombre_variable
        |error
        ;
nombre_variable ::= ID_VARIABLE otras_variables
                        ;
otras_variables ::= otra_variable
                        |
                        ;
```

```
otra_variable ::= otra_variable nombre_otra_variable
                | nombre_otra_variable
                ;
nombre_otra_variable ::= COMA id_variable
                       ;
id_variable ::= ID_VARIABLE
              |error
              ;
asignacion ::=      IGUAL tipos_datos fin_instruccion
              |fin_instruccion
              |error
              ;
tipos_datos ::= expr
              |GET_ELEMENT_BY_ID cont_element_by_id
              ;
cont_element_by_id ::= PAI ID_ELEMENT PAD
                     |error
                     ;
estruc_if ::=   if estruc_else_if
              ;
if ::= IF condicion cont_estruc_if
     ;
condicion ::=  PAI expr PAD
              |error
              ;
```

```
cont_estruc_if ::=      THEN instrucciones_para_if
                        |error
                        ;

estruc_else_if ::=  else_if estruc_else
                        |estruc_else
                        ;
else_if ::= else_if un_else_if
                |un_else_if
                ;
un_else_if ::= inicio_else_if condicion cont_estruc_if
                        ;
inicio_else_if ::= ELSE IF
                        ;
estruc_else ::= else
                        |
                        ;
else ::= inicio_else instrucciones_para_if
          ;
inicio_else ::= ELSE
                        ;
fin_instruccion_error ::= SEMI
                                |error
                                ;
```

```
funciones_especiales ::=  funciones_especiales_parametros fin_instruccion_error
                          |funciones_especiales_sin_parametros fin_instruccion_error
                          |funciones_especiales_sin_tipo_return fin_instruccion_error
                          ;
funciones_especiales_parametros ::= ASC condicion
                                    |DESC condicion
                                    |LETPAR_NUM condicion
                                    |LETIMPAR_NUM condicion
                                    |REVERSE condicion
                                    ;
funciones_especiales_sin_parametros ::= CARACTER_ALEATORIO parentesis
                                        |NUM_ALEATORIO parentesis
                                        ;
funciones_especiales_sin_tipo_return ::= ALERT_INFO condicion
                                         |EXIT parentesis
                                         |REDIRECT parentesis
                                        ;
estruc_repeat ::= REPEAT asig_decla_repeat huntil instrucciones_para_if
                  ;
asig_decla_repeat ::= PAI declaracion_asginacion_repeat PAD
                      |error
                      ;

declaracion_asginacion_repeat ::= declaracion_repeat asignacion_repeat
                                  ;
```

```
declaracion_repeat ::= tipo_dato ID_VARIABLE
                       |ID_VARIABLE
                       |error
                       ;
asignacion_repeat ::=      IGUAL expr
                       |error
                       ;


huntil ::= HUNTIL condicion
           |error
           ;
estruc_while ::= WHILE condicion thenwhile instrucciones_para_if
                 ;
thenwhile ::=  THENWHILE
               |error
               ;
insert ::= INSERT estruc_cont_insert fin_instruccion_error
           ;
estruc_cont_insert ::=      PAI cont_insert PAD
                       |error
                       ;
```

```
cont_insert ::= cont_insert COMA cont_insert
                |ID_ELEMENT
                |ID_VARIABLE
                |error
                ;

instrucciones_para_if ::= estruc_init
                          |insert
                          |funciones_especiales
                          ;
```

# Gramática del lenguaje de la función INSERT

## Gramática del analizador léxico:

Una cadena encerrado por comillas:
cadena = [\"]([^\"])*[\"]
Una cadena encerrado por comillas y que no acepta espacios en blanco:
cadena_sin_espacios = [\"]([^\"\r\n\t\f\ ])*[\"]
Comentario linea:
comentario_linea = "!!"[^\r\n\t\f]+
Comentario de bloque:

comentario_bloque = "<!--"([^\-]|"-"[^\-] |"--"[^\>])*"-->"
Palabras Reservadas sin importar si es mayúscula o minúscula:
C_SPAM = [cC][_][sS][pP][aA][mM]
C_INPUT =  [cC][_][iI][nN][pP][uU][tT]
C_TEXTAREA = [cC][_][tT][eE][xX][tT][aA][rR][eE][aA]
C_SELECT = [cC][_][sS][eE][lL][eE][cC][Tt]
C_OPTION = [cC][_][oO][pP][tT][iI][oO][nN]
C_DIV = [cC][_][dD][iI][vV]
C_IMG = [cC][_][iI][mM][gG]
C_BR = [cC][_][bB][rR]
C_BUTTON = [cC][_][bB][uU][tT][tT][oO][nN]
C_H1 = [cC][_][hH][1]
C_P = [cC][_][pP]

Lineas de terminacion de instrucciones:
lineTerminator = \r|\n|\r\n
Espacios en blanco:
whiteSpace  = {lineTerminator} | [ \t\f]

Palabras reservadas:
href
background
color
Font-size
Font-family

Text-align
type
id
name
cols
rows
class
src
width
height
alt
Onclick

Símbolos:
/
[
]
<
>
=

## **Gramática del analizador Sintáctico:**

Símbolos terminales:
MENQ, CORI, PARAMETRO, IGUAL,STRING, STRING_WS, CORD,MAQ, DIAG,C_SPAM,
C_INPUT,C_TEXTAREA,C_SELECT,C_OPTION,C_DIV,C_IMG,C_BR,C_BUTTON,C_H1,C_P,INVALID

Símbolos no terminales:
Inicio,c_etiqueta_g,etq_generica,opcional_fin_etiqueta,etiqueta_g,inicio_g,fin_etiqueta_g,cierre_g,etiqueta_br,
inicio_br,cierre_etiqueta,parametros_etiquetas,varios_parametro_etiqueta,parametro_etiqueta,inicio_param_etiq,string,
fin_param_etiq

Gramática:

inicio ::= etq_generica
|fin_etiqueta_g
|etiqueta_br
;

c_etiqueta_g ::= C_SPAM
|C_INPUT
|C_TEXTAREA}
|C_SELECT
|C_OPTION
|C_DIV
|C_IMG

```
                        |C_BUTTON
                        |C_H1
                        |C_P
                        ;

etq_generica ::= etiqueta_g:e0    opcional_fin_etiqueta
               ;

opcional_fin_etiqueta ::= fin_etiqueta_g
                        |
                        ;

etiqueta_g ::=       inicio_g:e0
                   parametros_etiquetas:e
                      ;

inicio_g ::= MENQ c_etiqueta_g
               ;
fin_etiqueta_g ::=      MENQ DIAG cierre_g
                        ;
cierre_g ::=     c_etiqueta_g MAQ
                 |error
                 ;

etiqueta_br ::= inicio_br cierre_etiqueta
```

```
                ;
inicio_br ::= MENQ C_BR
                ;

cierre_etiqueta ::= MAQ
                    |error
                    ;

parametros_etiquetas ::= varios_parametro_etiqueta MAQ
                         |MAQ
                         |error
                         ;

varios_parametro_etiqueta ::=    varios_parametro_etiqueta:e parametro_etiqueta:e1
                                |parametro_etiqueta:e
                                ;

parametro_etiqueta ::= inicio_param_etiq:e1 string:e2 fin_param_etiq
                       ;

inicio_param_etiq ::= CORI PARAMETRO:e IGUAL
                      ;

string ::=   STRING
             |STRING_WS
```

```
            |error
            ;
fin_param_etiq ::=      CORD
                        |error
                        ;
```

## Gramática de la Base de datos:

### Gramática del analizador léxico:

Una cadena de caracteres encerrado entre la sucesión de [$] :
cadena = "[$]"([^\[]|"["[^\$]|"[$"[^\]])*"[$]"
Lineas terminales:
lineTerminator = \r|\n|\r\n
Espacios en blanco:
whiteSpace = {lineTerminator} | [ \t\f]

Palabras Reservadas:
CAPTCHAS
CAPTCHA
ID
CODIGO_HTML
LINK

NOMBRE_CAPTCHA
CANT_UTILIZADOS
ACIERTOS
FALLOS
LAST_FECHA
SIMBOLOS
POSICION
IDENTIFICADOR
TIPO
VALOR
MODO
PROCEDIMIENTO
NO_EJECUCION
PROCESO_DECLARADO
PROCESO_DECLARADO_ESTRUCT

Símbolos:
{
}
(
)
=
:
;

**Gramática del analizador Sintáctico:**

Símbolos terminales:
CAPTCHAS, DOS_PUNTOS, PAI, PAD,CAPTCHA,LLAI,LLAD,ID, IGUAL, STRING,SEMI,CODIGO_HTML,LINK, NOMBRE_CAPTCHA,CANT_UTILIZADOS,ACIERTOS,FALLOS,LAST_FECHA,SIMBOLOS,POSICION, IDENTIFICADOR,TIPO,VALOR,MODO,PROCEDIMIENTO,N_E,P_D,P_D_E,INVALID

Símbolos no terminales:
inicio,sig_inicio,captchas,captcha,sig_captcha,cont_captcha,fin_instruccion,id,html,link,nombre_captcha,cant_u,aciertos, fallos,l_fecha,simbolos,cont_simbolos_lambda,cont_simbolos,simbolo,cont_sim,pos,iden,tipo,valor,modo,process,n_e, p_d,p_d_e

Gramática;

```
inicio ::= CAPTCHAS sig_inicio
           ;
sig_inicio ::= DOS_PUNTOS PAI captchas PAD
             |error
             ;

captchas ::=  captchas captcha
             |captcha
             ;

captcha ::= CAPTCHA sig_captcha
           ;
```

```
sig_captcha ::= DOS_PUNTOS LLAI cont_captcha LLAD
                |error
                ;

cont_captcha ::= id html link nombre_captcha cant_u: aciertos fallos l_fecha simbolos
                ;
fin_instruccion ::= SEMI
                    |error
                    ;

id ::= ID IGUAL STRING fin_instruccion
    ;

html ::= CODIGO_HTML IGUAL STRING fin_instruccion
        ;

link ::= LINK IGUAL STRING fin_instruccion
          ;

nombre_captcha ::= NOMBRE_CAPTCHA IGUAL STRING fin_instruccion
                               ;

cant_u ::= CANT_UTILIZADOS IGUAL STRING fin_instruccion
                ;
```

```
aciertos ::= ACIERTOS IGUAL STRING fin_instruccion    ;

fallos ::= FALLOS IGUAL STRING fin_instruccion    ;

l_fecha ::= LAST_FECHA IGUAL STRING fin_instruccion    ;

simbolos ::= SIMBOLOS IGUAL PAI cont_simbolos_lambda PAD
             |error
             ;

cont_simbolos_lambda ::= cont_simbolos
                         |
                         ;

cont_simbolos ::=      cont_simbolos simbolo
                      |simbolo
                      ;
simbolo ::= LLAI cont_sim LLAD
             |error
             ;
cont_sim ::= pos iden tipo valor modo process n_e p_d p_d_e
                ;
pos ::= POSICION IGUAL STRING:e fin_instruccion
        ;
```

```
iden ::= IDENTIFICADOR IGUAL STRING fin_instruccion
        ;
tipo ::= TIPO IGUAL STRING fin_instruccion
        ;
valor ::= VALOR IGUAL STRING fin_instruccion
        ;
modo ::= MODO IGUAL STRING fin_instruccion
         ;
process ::= PROCEDIMIENTO IGUAL STRING fin_instruccion
           ;
n_e ::= N_E IGUAL STRING fin_instruccion
        ;
p_d ::= P_D IGUAL STRING fin_instruccion
        ;
p_d_e ::= P_D_E IGUAL STRING fin_instruccion
          ;
```