

# 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 i7-7700HQ

## Tabla de Contenidos

Manual Técnico.....	1
Analizador Sintáctico.....	1
Gramática del lenguaje GCIC.....	2
Gramática de analizador léxico.....	2
Gramática para el analizador Sintactico:.....	7
Gramática del lenguaje de la función INSERT.....	25
Gramática del analizador léxico:.....	25
Gramática del analizador Sintáctico:.....	27
Gramática de la Base de datos:.....	31
Gramática del analizador léxico:.....	31
Gramática del analizador Sintáctico:.....	33

## 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\tf])\*\"

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

getElementById

ASC

DESC

LETPAR\_NUM  
LETIMPAR\_NUM  
REVERSE  
CHARACTER\_ALEATORIO  
NUM\_ALEATORIO  
ALERT\_INFO  
EXIT  
@global  
integer  
string  
boolean  
decimal  
char  
true  
false  
IF  
ELSE  
THEN  
INIT  
END  
REPEAT  
UNTIL  
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,  
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,estruc\_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 parenthesis
                    ;

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

funcion_process_ ::= name_funcion_g parenthesis
                    ;
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
                                       ;

estruc_repeat ::= REPEAT asig_decla_repeat huntill 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
                   ;

```

```

until ::= 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 línea:

```
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

$$\begin{array}{l} |C\_H1 \\ |C\_P \\ ; \end{array}$$

$$\text{etq\_generica} ::= \text{etiqueta\_g:e0} \quad \text{opcional\_fin\_etiqueta} \\ ;$$

$$\text{opcional\_fin\_etiqueta} ::= \text{fin\_etiqueta\_g} \\ | \\ ;$$

$$\text{etiqueta\_g} ::= \begin{array}{l} \text{inicio\_g:e0} \\ \text{parametros\_etiquetas:e} \\ ; \end{array}$$

$$\text{inicio\_g} ::= \text{MENQ c\_etiqueta\_g} \\ ;$$

$$\text{fin\_etiqueta\_g} ::= \text{MENQ DIAG cierre\_g} \\ ;$$

$$\text{cierre\_g} ::= \begin{array}{l} \text{c\_etiqueta\_g MAQ} \\ \text{error} \\ ; \end{array}$$

$$\text{etiqueta\_br} ::= \text{inicio\_br cierre\_etiqueta} \\ ;$$

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
      ;

```