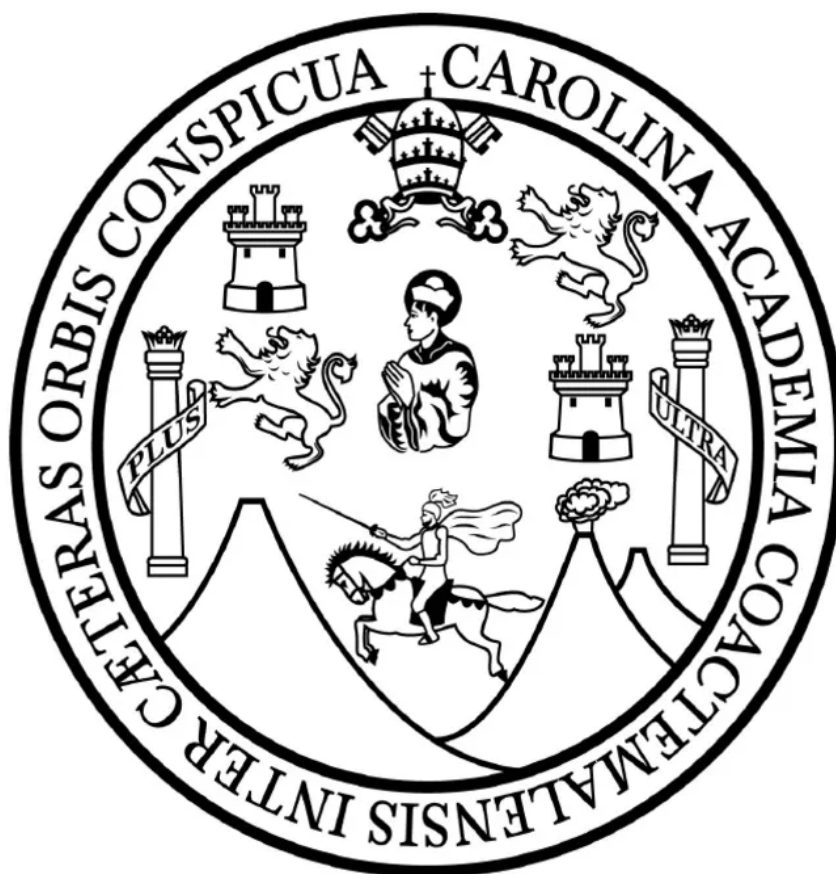


**Universidad de San Carlos de Guatemala**  
**Centro Universitario de Occidente**  
**División De Ciencias de la Ingeniería**



**Manual Tecnico Proyecto01**

**Rudy Adolfo Pacheco Pacheco**

**Ing. Jose Moises Granados Guevara**

# Análisis Léxico

## Java

Espacios = [\r\n|\r\n|\s\t] | [\t\f]

INICIOCOMENMULTILINEA = "/\*"

FINCOMENMULTILINEA = "\*/"

INICIOCOMENLINEA="//"

LETRA = [a-zA-Z|ñ]

DECIMAL = ([0-9]+[.]( [0-9]+))

ENTERO = [0-9]+

LLAVEA = "{"

LLAVEC = "}"

PARENTESISA = "("

PARENTESIS C = ")"

PUNTO = "."

COMA = ","

PUNTOCOMA = " ; "

DOPUNTO = " : "

COMILLAS = "\""

MENOS = "-"

MAS = "+"

MULTIPLICACION = "\*"

DIVISION = "/"

IGUAL="="

NEGATION = "!"

EQUALS={ {IGUAL} {IGUAL} }

OR="||"

AND="&&"

MENORQ="<"

MAYORQ=">"

MAYOROI=">="

MENOROI="<="

NOTEQUALS="!="

IMPORT="import"

PRIVATE = "private"

PUBLIC = "public"

PROTECTED = "protected"

FINAL = "final"

CLASS="class"

INT = "int"

BOOLEAN = "boolean"

STRING = "String"

CHAR = "char"

DOUBLE = "double"

IF = "if"

ELSE = "else"

FOR = "for"  
WHILE = "while"  
DO = "do"  
SWITCH = "switch"  
BREAK="break"  
RETURN = "return"  
NEW="new"  
DEFAULT = "default"  
CASO = "case"  
TRUE = "true"  
FLASE = "false"  
ID = {LETRA}{\_}({LETRA}{ENTERO}{\_})\*({LETRA}{ENTERO}}){LETRA}  
CARACTER = ""[^]""

## Json

ESPACIO = [\r\n\r\n\s\t] | [\t\f]  
PUNTO = "."  
DOSPUNTO = ":"  
COMA = ","  
PUNTOCOMA = ";"  
ENTERO = [0-9]+  
DECIMAL = ([0-9]+.[0-9]+)  
COMILLAS= "\""  
CORCHETEA= "["  
CORCHETEC= "]"  
LLAVEA = "{"  
LLAVEC = "}"  
LETRA = [a-zA-Zñ]  
SCORE="score"  
CLASES="clases"  
VARIBALES="variables"  
METODOS="metodos"  
COMENTARIOS="comentarios"  
FUNCION="funcion"  
TIPO="tipo"  
NOMBRE="nombre"  
PARAMTETROS="parametros"  
TEXTO="texto"  
ID = {LETRA}{\_}({LETRA}{ENTERO}{\_})\*({LETRA}{ENTERO}}){LETRA}

## Html

ESPACIO = [r|n|r|n|s|t] | [t|f]  
DECIMAL = ([0-9]+.[.][0-9]+))  
LETRA = [a-zA-Z|ñ]  
HTMLINICIO="<html>"  
HTMLFIN="</html>"  
H1INICIO="<h1>"  
H1CIERRE="</h1>"  
H2ABRE="<h2>"  
H2CIERRE="</h2>"  
TABLAABRE="<table>"  
TABLACIERRE="</table>"  
TRABRE="<tr>"  
TRCIERRE="</tr>"  
THABRE="<th>"  
THCIERRE="</th>"  
TDABRE="<td>"  
TDCIERRE="</td>"  
SALTO="<br>"  
FORABRE="<for"  
FORCIERRE="</for>"  
PUNTO = "."  
DOSPUNTO = ":"  
HASTA="hasta"  
ITERADOR = "iterador";  
MAYOR=">"  
MENOR="<"  
POSABERTURA = (({MENOR})({LETRA})+({MAYOR})?)  
POSCERRADURA = ((({MENOR})({DIVISION})({LETRA})+({MAYOR})))  
COMA = ","  
PUNTOCOMA = ","  
ENTERO = [0-9]+  
DIGONALB = " \_"  
ID = {LETRA}{\_}({LETRA}{ENTERO}{\_})\*({LETRA}{ENTERO}){LETRA}  
COMODIN="\$ \$"  
PARENTESISABRE="{"  
PARENTESISCIERRE="}"  
IGUAL="="  
MENOS = "-"  
MAS = "+"  
POR = "\*"  
DIVISION = "/"  
COMILLAS="\"  
CORCHETEA="[ "  
CORCHETEC="]"

```
INICIO_COMENTARIO_MULTILINEA = "</"
FIN_COMENTARIO_MULTILINEA = ">"
INTEGER="integer"
STRING="string"
```

# Analisis Sintactico

## Java

### No terminales

inicio  
item\_punto  
importando  
definicion\_clase  
definicion\_metodo  
definicion\_variable  
variables  
items\_coma  
items  
fin\_asignar  
asignacion  
tipo  
visibilidad  
sentencias  
sentencia  
sentencias\_global  
sentencia\_global  
definicion\_if  
definicion\_ifelse  
definicion\_for  
acceso\_termin\_var\_metod  
definicion\_while  
definicion\_dowhile  
definicion\_switch  
cases  
caso\_sw  
factor\_casos  
terminal\_casos  
salid  
retorno  
condition  
logica  
sentencias  
switch  
agrupation  
definicion\_and

definicion\_or  
condition\_for  
operation\_for  
constructor  
parametros  
llamada\_metodo  
parametros\_llamada  
recurcion\_new  
increment  
mas\_igual  
var\_llamad\_asigan  
idOption  
siguiente  
var\_iterador  
e\_inicial  
t\_production  
e\_pri  
t\_pri  
f\_terminal

## Terminales

ENTERO  
DECIMAL  
LLAVEA  
LLAVEC  
PARENTESISA  
PARENTESIS C  
PUNTO  
COMA  
PUNTOCOMA  
DEFAULT  
DOSPUNTOS  
MENOS  
MAS  
MULTIPLICACION  
DIVISION  
IGUAL  
EQUALS  
OR  
AND  
MENORQ  
MAYORQ  
MAYOROI  
MENOROI  
CASO  
NOTEQUALS

IMPORT  
ID  
PRIVATE  
PUBLIC  
PROTECTED  
FINAL  
CLASS  
INT  
BOOLEAN  
TRUE  
CADENA  
STRING  
NEGATION  
CHAR  
CHARACTER  
DOUBLE  
IF  
ELSE  
FOR  
WHILE  
DO  
SWITCH  
BREAK  
RETURN  
NEW  
FALSE

gramática para las importaciones

importando -> IMPORT ID PUNTO item\_punto  
| IMPORT ID PUNTOCOMA importando  
| error PUNTOCOMA

item\_punto -> ID PUNTO item\_punto  
| ID PUNTOCOMA importando  
| MULTIPLICACION PUNTOCOMA importando  
| error PUNTOCOMA

//Definicion de clase

definicion\_clase -> visibilidad comodin\_Class LLAVEA sentencias\_global LLAVEC  
| FINAL comodin\_Class LLAVEA sentencias\_global LLAVEC  
| comodin\_Class LLAVEA sentencias\_global LLAVEC

comodin\_Class -> CLASS ID

//definicion de clase o metodo

definicion\_metodo -> visibilidad comodin\_Metod LLAVEA sentencias LLAVEC

| error LLAVEC {::}

;

comodin\_Metodo -> tipo ID:nomMet PARENTESISA parametros PARENTESISC

| tipo ID:nomMet PARENTESISA PARENTESISC

| ID:tipMetodo ID:nomMet PARENTESISA PARENTESISC

| ID:tipMetodo ID:nomMet PARENTESISA parametros PARENTESISC

//definicion de variable o asignacion

definicion\_variable -> visibilidad variables

| variables

| FINAL variables

| visibilidad ID:tipoVar items\_coma

| FINAL ID:tipoVar items\_coma

| ID:tipoVar items\_coma

variables -> tipo items\_coma

//items consecutivos

items\_coma -> ID:nomVar items

| error PUNTOCOMA

//varias asignaciones

items -> IGUAL asignacion PUNTOCOMA

| IGUAL asignacion COMA items\_coma

| COMA items\_coma

| PUNTOCOMA

//palabra final

fin\_asignar -> FINAL

//asignacion

asignacion -> condition



```
| NEW ID PARENTESIS parametros_llamada PARENTESIS  
| NEW ID PARENTESIS PARENTESIS
```

```
//tipo
```

```
tipo -> INT  
| BOOLEAN  
| DOUBLE  
| STRING  
| CHAR  
;
```

```
//visibilidad
```

```
visibilidad ->:PUBLIC fin_asignar  
| PROTECTED fin_asignar  
| PRIVATE fin_asignar  
;
```

```
// sentencias
```

```
sentencias ->sentencia sentencias  
|  
| error sentencias  
;
```

```
//globales
```

```
sentencias_global -> sentencia_global sentencias_global  
|  
;
```

```
sentencia_global *> constructor
```

```
| definicion_variable  
| definicion_metodo  
;
```

```
//sentencias
```

```
sentencia -> definicion_if  
| definicion_ifelse  
| definicion_while  
| definicion_dowhile  
| definicion_for  
| FINAL variables  
| salid  
| variables  
| ID idOption  
| ID:tipoVar items_coma  
| definicion_switch
```

| FINAL ID:tipoVar items\_coma

idOption ->increment

| var\_llamad\_asigan mas\_igual asignacion PUNTOCOMA

| PARENTESIS PARENTESIS var\_llamad\_asigan mas\_igual asignacion

PUNTOCOMA

| PARENTESIS parametros\_llamada PARENTESIS var\_llamad\_asigan mas\_igual

asignacion PUNTOCOMA

| mas\_igual asignacion PUNTOCOMA

| llamada\_metodo PUNTOCOMA

| PARENTESIS PARENTESIS PUNTOCOMA

| PARENTESIS parametros\_llamada PARENTESIS PUNTOCOMA

| PARENTESIS PARENTESIS llamada\_metodo PUNTOCOMA

| PARENTESIS parametros\_llamada PARENTESIS llamada\_metodo

PUNTOCOMA

var\_llamad\_asigan -> PUNTO ID PARENTESIS PARENTESIS var\_llamad\_asigan

| PUNTO ID PARENTESIS parametros\_llamada PARENTESIS

var\_llamad\_asigan

| PUNTO ID var\_llamad\_asigan

| PUNTO ID

mas\_igual -> IGUAL

| MAS IGUAL

;

increment -> MAS MAS PUNTOCOMA

| MENOS MENOS PUNTOCOMA

| var\_llamad\_asigan MAS MAS PUNTOCOMA

| var\_llamad\_asigan MENOS MENOS PUNTOCOMA

;

//if

definicion\_if -> IF PARENTESIS condition PARENTESIS LLAVEA sentencias LLAVEC

;

//if else

definicion\_ifelse -> definicion\_if ELSE LLAVEA sentencias LLAVEC

;

//for

definicion\_for -> FOR PARENTESIS condition\_for PARENTESIS LLAVEA sentencias LLAVEC

;

```
//while
definicion_while -> WHILE PARENTESISIA condition PARENTESISC LLAVEA sentencias
LLAVEC
;
```

```
//do while
definicion_dowhile -> DO LLAVEA sentencias LLAVEC WHILE PARENTESISIA condition
PARENTESISC PUNTOCOMA
;
```

```
//switch
definicion_switch -> SWITCH PARENTESISIA ID PARENTESISC LLAVEA cases LLAVEC
;
```

```
cases -> caso_switch solid cases
| caso_switch cases
```

```
caso_switch -> CASO agrupation DOPUNTO sentencias_switch
| DEFAULT DOPUNTO sentencias_switch
```

```
sentencias_switch -> factor_casos sentencias_switch
```

```
factor_casos -> definicion_if
| definicion_ifelse
| definicion_while
| definicion_dowhile
| definicion_for
| FINAL variables
| variables
| ID idOption
| ID:tipoVar items_coma
| definicion_switch
| FINAL ID:tipoVar items_coma
```

```
terminal_casos -> ID
| ID acceso_termin_var_metod
| ID PARENTESISIA PARENTESISC
| ID PARENTESISIA parametros_llamada PARENTESISC
| ID PARENTESISIA PARENTESISC acceso_termin_var_metod
| ID PARENTESISIA parametros_llamada PARENTESISC
acceso_termin_var_metod
| ENTERO
```

```
| DECIMAL
| CHARACTER
| CADENA
;
```

```
acceso_termin_var_metod -> PUNTO ID PARENTESIS PARENTESIS
acceso_termin_var_metod
    | PUNTO ID PARENTESIS parametros_llamada PARENTESIS
acceso_termin_var_metod
    | PUNTO ID acceso_termin_var_metod
    | PUNTO ID
    | PUNTO ID PARENTESIS parametros_llamada PARENTESIS
    | PUNTO ID PARENTESIS PARENTESIS
    | error PARENTESIS {::}
;
```

```
//punto y coma
salid -> BREAK PUNTOCOMA
    | RETURN retorno PUNTOCOMA
```

```
//return
retorno -> condition
```

```
condition -> logica
    | definicion_and
    | definicion_or
    | NEGATION definicion_and
    | NEGATION definicion_or
    | NEGATION logica
```

```
logica -> agrupation MAYORQ agrupation
    | agrupation MENORQ agrupation
    | agrupation MAYOROI agrupation
    | agrupation MENOROI agrupation
    | agrupation NOTEQUALS agrupation
    | agrupation EQUALS agrupation
    | TRUE
    | FALSE
    | e_inicial
    | CADENA PUNTO PARENTESIS PARENTESIS
    | CADENA PUNTO PARENTESIS parametros_llamada PARENTESIS
```

```
agrupation -> e_inicial
```

e\_inicial -> t\_production e\_pri

e\_pri -> MAS t\_production e\_pri  
| MENOS t\_production e\_pri

t\_production -> f\_terminal t\_pri  
;

t\_pri -> MULTIPLICACION f\_terminal t\_pri  
| DIVISION f\_terminal t\_pri  
;

f\_terminal -> PARENTESIS e\_inicial PARENTESIS  
| terminal\_casos

definicion\_and -> logica AND logica  
| logica AND logica AND condition  
| logica AND logica OR condition

definicion\_or -> logica OR logica  
| logica OR logica OR condition  
| logica OR logica AND condition

condition\_for -> var\_iterador ID IGUAL ID PUNTOCOMA condition PUNTOCOMA  
operation\_for  
| var\_iterador ID IGUAL ENTERO PUNTOCOMA condition PUNTOCOMA  
operation\_for  
| var\_iterador ID IGUAL DECIMAL PUNTOCOMA condition PUNTOCOMA  
operation\_for

var\_iterador -> INT  
| DOUBLE

operation\_for -> ID MAS MAS  
| ID MENOS MENOS  
| ID IGUAL condition

| ID MAS IGUAL condition  
| ID MENOS IGUAL condition

constructor-> visibilidad comodin\_Consturc LLAVEA sentencias LLAVEC  
| comodin\_Consturc LLAVEA sentencias LLAVEC

comodin\_Consturc -> ID:nomMet PARENTESISIA parametros PARENTESISISC  
| ID:nomMet PARENTESISIA PARENTESISISC

//parametros  
parametros -> comodin\_param  
| comodin\_param COMA parametros

comodin\_param -> tipo ID:nomPar  
| ID:tipoPar ID:nomPar

//llamada funciones  
llamada\_metodo -> PUNTO ID PARENTESISIA PARENTESISISC llamada\_metodo  
| PUNTO ID PARENTESISIA parametros\_llamada PARENTESISISC  
llamada\_metodo  
| PUNTO ID llamada\_metodo  
| PUNTO ID PARENTESISIA parametros\_llamada PARENTESISISC  
| PUNTO ID PARENTESISIA PARENTESISISC

/\*gramatica para los parametros de las llamadas\*/  
parametros\_llamada -> condition COMA parametros\_llamada  
| condition  
| NEW recursion\_new

recursion\_new ->ID PARENTESISIA parametros\_llamada PARENTESISISC siguiente  
| ID PARENTESISIA PARENTESISISC siguiente

siguiente -> COMA parametros\_llamada

## Json

### Terminales

SCORE  
CLASES  
VARIBALES  
METODOS  
COMENTARIOS  
FUNCION  
TIPO  
NOMBRE  
PARAMTETROS  
TEXTO  
ID  
DOSPUNTO  
COMA  
ENTERO  
DECIMAL  
COMILLAS  
CORCHETEA  
CORCHETEC  
LLAVEA  
LLAVEC

### No terminales

inicio  
definicion\_clase  
clases  
clase  
definicion\_score  
definicion\_variable  
variable, variables  
nombre\_var  
tipo\_Var  
funciones  
definicion\_comentario  
comodinComentario  
metodo;  
funcion\_padre  
componentes  
funcion\_oblig  
funcion\_Items  
definicion\_metodo

metods, parametros  
nom\_metodo  
tipo\_metodo  
idd\_Coment  
text  
coments  
comodinItem

inicio -> LLAVEA componentes LLAVEC ;

componentes -> definicion\_score definicion\_clase definicion\_variable definicion\_metodo  
def\_coment  
;

definicion\_score -> SCORE DOSPUNTO COMILLAS ENTERO:score COMILLAS COMA  
| SCORE DOSPUNTO COMILLAS DECIMAL:score COMILLAS COMA

definicion\_clase -> CLASES DOSPUNTO CORCHETEA clases CORCHETEC COMA  
| CLASES DOSPUNTO CORCHETEA CORCHETEC COMA

clases -> LLAVEA clase LLAVEC COMA clases  
| LLAVEA clase LLAVEC

clase -> NOMBRE DOSPUNTO COMILLAS IDD:nomClase COMILLAS

definicion\_variable -> VARIBALES DOSPUNTO CORCHETEA variables CORCHETEC  
COMA  
| VARIBALES DOSPUNTO CORCHETEA CORCHETEC COMA

variables -> LLAVEA variable LLAVEC COMA variables  
| LLAVEA variable LLAVEC

variable -> nombre\_var COMA tipo\_Var COMA funciones

nombre\_var -> NOMBRE DOSPUNTO COMILLAS ID COMILLAS

tipo\_Var -> TIPO DOSPUNTO COMILLAS IDD COMILLAS

funciones -> FUNCION DOSPUNTO COMILLAS funcion\_padre COMILLAS

funcion\_padre -> funcion\_oblig cion  
;

funcion\_oblig -> IDD:tipo1 IDD:funPadre1 COMA IDD IDD



| IDD:fun COMA IDD  
| IDD:funPadre1 COMA IDD IDD  
| ID

cion -> comodinItem cion

comodinItem -> COMA IDD:tipo1 IDD:funcionP {funcionPadre+=  
", "+tipo1.getLexeme()+" "+funcionP.getLexeme();}  
| COMA IDD:funcionP {funcionPadre+= ", "+  
funcionP.getLexeme();}  
;

definicion\_metodo -> METODOS DOSPUNTO CORCHETEA methods CORCHETEC  
COMA  
| METODOS DOSPUNTO CORCHETEA CORCHETEC COMA

methods -> LLAVEA metodo LLAVEC COMA methods  
| LLAVEA metodo LLAVEC

metodo-> nom\_Metod COMA tipo\_Metod COMA parametros

nom\_Metod -> NOMBRE DOSPUNTO COMILLAS IDDCOMILLAS

tipo\_Metod -> TIPO DOSPUNTO COMILLAS IDD:tipoMetod COMILLAS

parametros -> PARAMTETROS DOSPUNTO ENTERO

def\_coment -> COMENTARIOS DOSPUNTO CORCHETEA coments CORCHETEC  
| COMENTARIOS DOSPUNTO CORCHETEA CORCHETEC

coments -> LLAVEA text LLAVEC COMA coments  
| LLAVEA text LLAVEC

text -> TEXTO DOSPUNTO COMILLAS idd\_Coment COMILLAS

idd\_Coment -> comodinComentario idd\_Coment  
| comodinComentario

comodinComentario -> IDD  
| ALGO  
| COMA

Html

## Terminales

HTMLINICIO  
ENTERO  
DECIMAL  
HTMLFIN  
H1INICIO  
H1FIN  
H2INICIO  
H2FIN  
TABLAINICIO  
TABLAFIN  
TRINICIO  
TRFIN  
THINICIO  
THFIN  
TDINICIO  
TDFIN  
SALTO  
FORINICIO  
FORFIN  
PUNTO  
DOSPUNTO  
HASTA  
MAYOR  
MENOR  
POSABERTURA  
POSCERRADURA  
COMA  
PUNTOCOMA  
ID  
COMODIN  
PARENTESISA  
PARENTESIS  
IGUAL  
MENOS  
MAS  
POR  
DIVISION  
CORCHETEA  
CORCHETEC  
INTEGER  
STRING  
SCORE  
CADENA

RESULT  
VARIABLES  
CLASES  
NOMBRE  
TIPO  
FUNCION  
METODOS  
COMENTARIOS  
TEXTO  
PARAMETROS  
ITERADOR

## No terminales

def\_variable  
variable\_integer  
variable\_String  
string  
items\_coma  
items  
asignacion  
idVarComodin  
titleRescursiver  
inte  
items\_int  
items\_coma\_B  
asignacionInt  
inicio  
var\_Asign  
idItemsComodi  
iddComodinInt  
operationaux  
sentenciasH1H2  
operationString  
concatenacion  
subTerminalTabla  
def\_var\_Globla  
terminalTabla  
operationAsig  
index  
parametros  
operation  
ltitulo1  
itemHtml  
comodinIddHtml  
usoVarGlobal  
titulo2

columnTitle  
comodinH1H2  
columnDatos  
fila\_columna  
fila\_columnaRecur  
validosTR  
forAnidado  
comodintitle  
definicion\_table  
comodinTable  
definicion\_html\_General  
sentencias  
sentencia  
definicion\_for  
sentenciaFor

inicio ::= definicion\_variable definicion\_html\_General  
;

definicion\_variable ::= variable\_integer definicion\_variable  
| variable\_String definicion\_variable  
| var\_Asign definicion\_variable

variable\_String ::= strin items

strin -> STRING IDD:identificador

items\_coma -> iddItemsComodin items

iddItemsComodin -> ID:nomVar

items -> asignacion PUNTOCOMA  
| asignacion COMA items\_coma  
| COMA items\_coma  
| PUNTOCOMA

asignacion -> IGUAL operationString:contenido

operationString -> CADENA:cadena MAS concatenacion:contenido  
| CADENA:contenido  
| subTerminalTabla:contenido MAS concatenacion:contenido1

```

| subTerminalTabla:contenido
| IDD:identificador
| IDD:identificador MAS concatenacion:contenido ;

```

concatenacion ::= concatenacion MAS concatenacion:contenido2

```

| ENTERO:contenido
| DECIMAL:contenido
| terminalTabla:contenido
| CADENA:contenido
| IDD:identificador

```

variable\_integer ::= inte items\_int

inte -> INTEGER ID

```

items_int -> asignacionInt PUNTOCOMA
| asignacionInt COMA items_coma_B
| COMA items_coma_B
| PUNTOCOMA

```

items\_coma\_B \*> iddComodinInt items\_int

iddComodinInt -> IDD:identificador

asignacionInt -> IGUAL operation:valOperation

```

operation -> operation:numero1 MAS:operador operation:numero2
| operation:numero1 MENOS:operador operation:numero2
| operation:numero1 POR:operador operation:numero2
| operation:numero1 DIVISION:operador operation:numero2
| ENTERO:num

```

```

| DECIMAL:
| parametros
| IDD:identificador
| PARENTESIS operation:operacion PARENTESIS

```

subTerminalTabla -> RESULT PUNTO definicion\_var\_Globla:contenido

;

```

definicion_var_Globla -> SCORE:score
| CLASES CORCHETEA:pos index:indice CORCHETEC PUNTO NOMBRE
| VARIABLES CORCHETEA:pos index:indice CORCHETEC PUNTO NOMBRE

```

- | VARIABLES CORCHETEA:pos index:indice CORCHETEC PUNTO TIPO
- | VARIABLES CORCHETEA:pos index:indice CORCHETEC PUNTO FUNCION
- | METODOS CORCHETEA:pos index:indice CORCHETEC PUNTO NOMBRE
- | METODOS CORCHETEA:pos index:indice CORCHETEC PUNTO TIPO
- | COMENTARIOS CORCHETEA:pos index:indice CORCHETEC PUNTO TEXTO

terminalTabla -> RESULT PUNTO definicion\_var\_Globla:contenido

- | RESULT PUNTO parametros:contenido

parametros -> METODOS CORCHETEA:pos index:indice CORCHETEC PUNTO  
PARAMTETROS

index -> ENTERO:numero

- |IDD: numero ;

var\_Asign -> iddVarComodin IGUAL operationAsig:contenido PUNTOCOMA

iddVarComodin -> ID:identificador

- ;

operationAsig ::= operationAsig:numero1 MAS:operador operationAsig:numero2

- | operationAsig:numero1 MENOS:operador operationAsig:numero2

- | operationAsig:numero1 POR:operador operationAsig:numero2

- | operationAsig:numero1 DIVISION:operador operationAsig:numero2

- | ENTERO:contenido

- | DECIMAL:contenido

- | terminalTabla:contenido

- | CADENA:contenido

- | ID:num

- | operationaux

operationaux ::= PARENTESIS operation:operacion PARENTESIS

titulo1 -> H1INICIO itemHtml H1FIN

titleRescursiver ::= comodintitle titleRescursiver

- | comodintitle

- ;

itemHtml ::= comodinIddHtml itemHtml

- | usoVarGlobal itemHtml

- |

- ;

comodinIddHtml ::= ID:contenido

- | CADENA:contenido

```

| ENTERO:contenido
| DECIMAL:contenido
| COMA:contenido
| PUNTOCOMA:contenido
| PARENTESISIA:contenido
| PARENTESISIC:contenido
| IGUAL: contenido
| MENOS: contenido
| MAS: contenido
| POR: contenido
| DIVISION:contenido
| CORCHETEA: contenido
| CORCHETEC: contenido
| INTEGER: contenido
| STRING:contenido
| PUNTO:contenido
| DOSPUNTO:contenido
| HASTA:contenido
| MAYOR:contenido
| MENOR:contenido
;

```

```

usoVarGlobal ::= COMODIN PARENTESISIA IDD:identificador PARENTESISIC COMODIN
| COMODIN PARENTESISIA terminalTabla:contenido PARENTESISIC COMODIN
;

```

```

titulo2 ::= H2INICIO itemHtml H2FIN
;

```

```

columnTitle ::= THINICIO titleRescursiver THFIN
| THINICIO THFIN
;

```

```

comodinH1H2 ::= titulo1
| titulo2
| SALTO
| comodinIddHtml
| usoVarGlobal
;

```

```

comodintitle ::= titulo1
| titulo2
| SALTO
| comodinIddHtml
| usoVarGlobal
;

```

```
columnDatos -> TDINICIO titleRescursiver TDFIN  
| TDINICIO TDFIN
```

```
fila_columna -> TRINICIO validosTR TRFIN  
| TRINICIO TRFIN
```

```
fila_columnaRecur ::= fila_columna fila_columnaRecur  
| fila_columna
```

```
validosTR-> columnDatos validosTR  
| columTitle validosTR  
| columnDatos  
| columTitle  
| comodintitle validosTR  
| comodintitle
```

```
forAnidado ::= FORINICIO ITERADOR DOSPUNTO IDD HASTA DOSPUNTO IDD  
PUNTOCOMA MAYOR fila_columnaRecur FORFIN  
| FORINICIO ITERADOR DOSPUNTO IDD HASTA DOSPUNTO IDD PUNTOCOMA  
MAYOR FORFIN  
;
```

```
definicion_tabla ::= TABLAINICIO comodinTable TABLAFIN  
| TABLAINICIO TABLAFIN ;
```

```
comodinTable ::= fila_columna comodinTable  
| comodintitle comodinTable  
| fila_columna  
| comodintitle  
| forAnidado  
| forAnidado comodinTable  
;
```

```
definicion_html_General ::= HTMLINICIO sentencias HTMLFIN  
{:actionSalid.toString();:}  
| HTMLINICIO HTMLFIN  
;
```

```
sentencias ::= sentencia sentencias  
| sentencia  
;
```

```
sentencia ::= comodinH1H2  
| definicion_tabla  
| def_for
```



;

def\_for ::= FORINICIO ITERADOR DOSPUNTO IDD HASTA DOSPUNTO IDD  
PUNTOCOMA MAYOR sentenciasFro FORFIN  
| FORINICIO ITERADOR DOSPUNTO IDD HASTA DOSPUNTO IDD PUNTOCOMA  
MAYOR FORFIN

;

sentenciasFro ::= sentenciaFor sentenciasFro  
| sentenciaFor  
;

sentenciaFor ::= comodinH1H2  
| definicion\_tabla  
;