

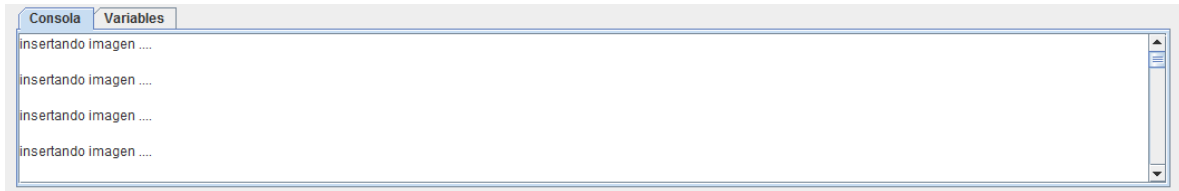
MANUAL TECNICO

Manual para usuario avanzado describiendo el generador de paginas web
usando Struct y Hs

201700886

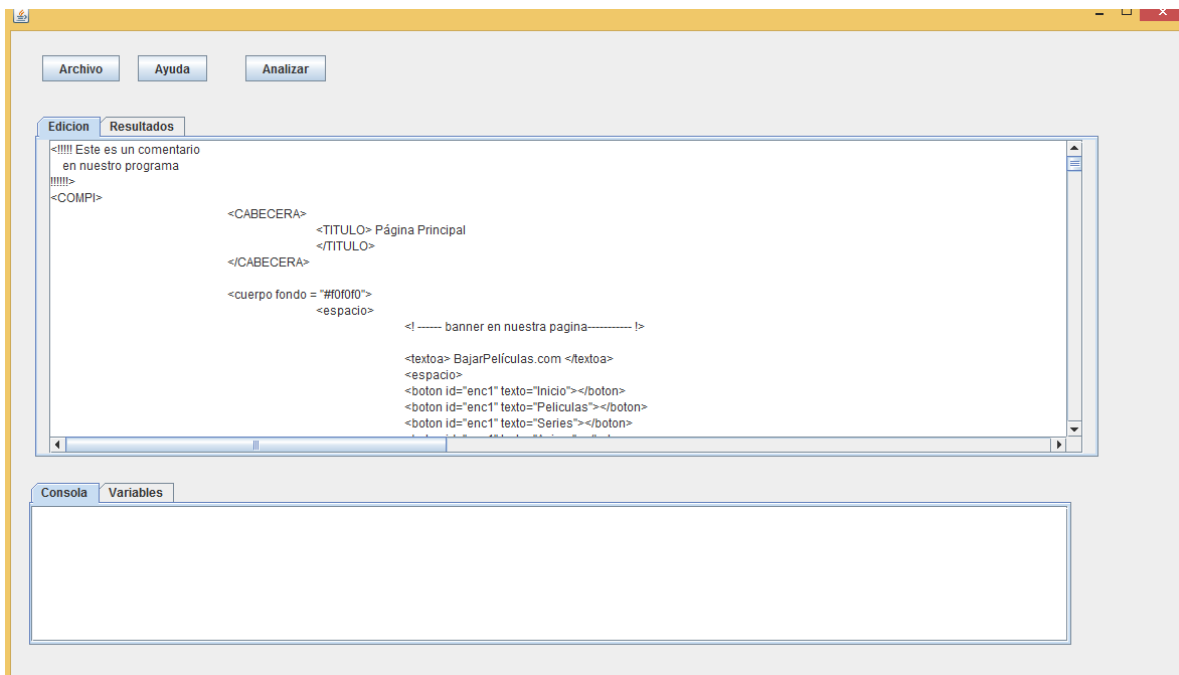
Error Léxico:

Al encontrar un símbolo que no pertenezca al lenguaje se informara el carácter erróneo y su ubicación.



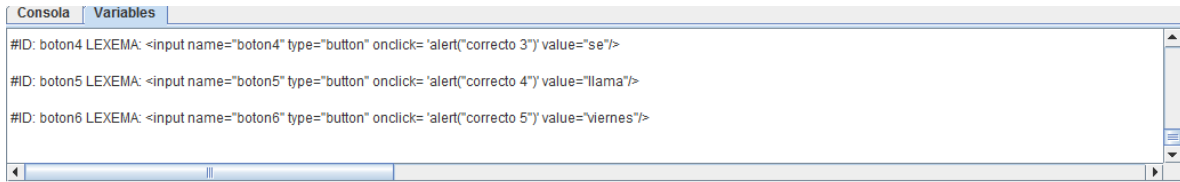
Interfaz Principal

Interfaz que se muestra al abrir por primera vez el programa



Error Sintáctico:

El graficador al no encontrar o encontrar un símbolo informara al usuario, al momento de encontrar el error no realizara nada más que notificar del mismo



Lista De Símbolos Terminales

- entero,decimal,cadena,booleano,id,caracter, puntocomma;
- suma, resta, multi, div, para, parc,esp,interro;
- may,men,admira,salto,alineacion;
- comen,compi,cabecera,titulo,cuerpo,fondo,igual,color,rgb,espacio;
- parrafo,izquierda,derecha,centrado,justificado,alip,textoa,textob;
- imagen,path,imgdir,alto,ancho,boton,idboton,textoboton,tabla,borde,estado;
- fila,columnac,columna,hs,dolar,echo;
- punto,ifv,anddoble,ordoble,llaa,llac,cora,corc,and,or,menorigual,mayorigual,diferente,igualdoble;
- repetir,numeral,insertar,crearparrafo,setalineacion,getalineacion,setcontenido,getcontenido;
- creartextoa,creatextob,crearimagen,setpath,getpath,setalto,getalto,setancho,getancho ;
- setborder,crearboton,setttexto,getttexto,clickboton,elses,coma, especialista,creattabla ;

Lista De Símbolos No Terminales

- STRUCT;
- COMENTA, CABE,CUERP,CONTE,CONTES,CONTESS,CO,COL,ESPECIAL;
- ESPA,ESPAP,PARRA,PARRAP,ALI,ALIP,TEXA,TEXB,IMA,SAL;
- IMAP,IMAPP,BOT,BOTP,TAB,FIL,COLU,MCON,ALT1,ALT2;
- SCRIPT,CUERS,VAR,VAL,VALN,ECH,CONCA,CONCAS,CONCASS,ANUM,MCONN;
- SI,ELS,INTE,COMPA;
- REP,ANUMP,CUERSP,CUERSP2;
- NATIV,STRH,MH,MHS,VAL1,VALN1;
- TABA,NBO,MUC,FUBOT,MCONNN,ATEMPO,ATEMPO2,STRM,PZ,MUCHO,IMD;

Gramática De tipo 2 para el analizador Sintáctico

TABA::= numeral id:h igual creartabla para MUCHO:a parc puntocomas { : MO.AF(h,5);
MO.FSC(h,a); RESULT= "";; }

| numeral id:z punto setborder para NBO:a parc puntocomas { :MO.FSB(z,a);
RESULT="";; }

;

NBO::= dolar id:a { :RESULT=MO.R(a,"\$"); }

| booleano:a { :RESULT=a;; }

;

MUCHO::= cora MUC:a corc coma MUCHO:b { :RESULT="<TR>" + a + "</TR>" + b;; }

| cora MUC:a corc { :RESULT="<TR>" + a + "</TR>";; }

;

MUC::= VAL:a coma MUC:b { :RESULT="<TD>" + a + "</TD>" + b;; }

| VAL:a { :RESULT="<TD>" + a + "</TD>";; }

;

FUBOT::= numeral id:z igual crearboton para STRM:a coma STRM:b parc puntocomas
{ :MO.AF(z,6); MO.FSTB(z,b); MO.FSAB(z,a); RESULT="";; }

|numeral id:z punto settexto para STRM:a parc puntocoma { : MO.FSTB(z,a);
RESULT="";; }

|dolar id:z igual numeral id:b punto gettexto para parc puntocoma
{ :MO.A(z,MO.FGTB(b),"\$"); RESULT="";; }

|numeral id:z punto clickboton para STRM:b parc puntocoma { : MO.FSAB(z,b);
RESULT="";; }

;

//////////PAGINA 7

NATIV::=numeral id:a punto insertar para parc puntocoma { :RESULT=MO.Insertar(a);: }

|numeral id:a igual crearparrafo { :MO.AF(a,1);Funci=a; : } para MH:c parc
puntocoma { : Funci="" ; RESULT="";; }

|numeral id:a punto setalineacion para ALIP:c parc puntocoma { : MO.FSA(a,c);
Funci="" ; RESULT="";; }

|dolar id:a igual numeral id:b punto getalineacion para parc puntocoma
{ :MO.A(a,MO.FGA(b),"\$"); Funci="" ; RESULT="";; }

|numeral id:a punto setcontenido para STRM:c parc puntocoma { :MO.FSC(a,c);
Funci="" ; RESULT="";; }

|dolar id:a igual numeral id:b punto getcontenido para parc puntocoma
{ :MO.A(a,MO.FGC(b),"\$"); Funci="" ; RESULT="";; }

|numeral id:a igual creartextoa para STRM:c parc puntocoma
{ :MO.AF(a,2);MO.FSC(a,c); Funci="" ; RESULT="";; }

|numeral id:a igual creartextob para STRM:c parc puntocoma
{ :MO.AF(a,3);MO.FSC(a,c); Funci="" ; RESULT="";; }

|numeral id:a igual crearimagen { :MO.AF(a,4);Funci=a; : } para PZ:c parc puntocoma
{ :Funci="" ; RESULT="";; }

|numeral id:a punto setpath para STRH:c parc puntocoma { :MO.FSP(a,c); Funci="" ;
RESULT="";; }

|dolar id:a igual numeral id:b punto getpath para parc puntocoma
{ :MO.A(a,MO.FGP(b),"\$"); Funci="" ; RESULT="";; }

|numeral id:a punto setalto para VALN:c parc puntocomma { :MO.FSAL(a,c);Funci="";
RESULT="";: }

|dolar id:a igual numeral id:b punto getalto para parc puntocomma
{ :MO.A(a,MO.FGAL(b),"\$"); Funci=""; RESULT="";: }

|numeral id:a punto setancho para VALN:c parc puntocomma { :MO.FSAN(a,c);
Funci=""; RESULT="";: }

|dolar id:a igual numeral id:b punto getancho para parc puntocomma
{ :MO.A(a,MO.FGAN(b),"\$"); Funci=""; RESULT="";: }

;

STRH::= imgdir:a { :RESULT=a;: }

| cadena:a { :RESULT=a;: }

|dolar id:a { :RESULT="\$"+a;: }

;

STRM::= cadena:a { :RESULT=a;: }

|numeral id:b punto gettexto para parc { : RESULT=MO.FGTB(b);: }

|dolar id:a { :RESULT=MO.R(a,"\$");: }

;

MH::= STRM:a coma ALIP:b { :MO.FSC(Funci, a); MO.FSA(Funci,b); RESULT=a+b;: }

|STRM:a { :MO.FSC(Funci,a); RESULT=a;: }

;

PZ::=STRH:a { : MO.FSP(Funci,a); RESULT=a;: }

|STRH:a coma VALN:b coma VALN:c { : MO.FSP(Funci,a); MO.FSAL(Funci,b);
MO.FSAN(Funci,c); RESULT=a+b+c;: }

|STRH:a coma coma VALN:b { :MO.FSP(Funci,a); MO.FSAN(Funci,b);
RESULT=a+b;: }

|STRH:a coma VALN:b coma { :MO.FSP(Funci,a); MO.FSAL(Funci,b);
RESULT=a+b;: }

;

/////////PAGINA 6

```

REP::=repetir para VAL:a { :NV=NV*MO.Pint(a); : } parc llaa CUERSP:b
{ :NV=NV/MO.Pint(a); String M=""; int lol=MO.Pint(a); for(int i=0;i<lol;i++){M=M+b;}
b=M; int Tamaño=NVeces.size(); RESULT=b;};

```

```

CUERSP::=ECH:a CUERSP:b { :RESULT=a+b;:}

```

```

|SI:a CUERSP:b { :RESULT=a+b;:}

```

```

|REP:a CUERSP:b { :RESULT=a+b;:}

```

```

|VAR:a CUERSP:b { :RESULT=a+b;:}

```

```

|numeral id:a punto insertar para parc puntocoma CUERSP:b
{ :RESULT=MO.Insertar(a)+b;:}

```

```

|llac { :RESULT="";:}

```

```

;

```

```

/////Pagina5          cadena+cadena1+cadena2;

```

```

SI::=ifv para INTE:cadena

```

```

{ :if(cadena.contains("true")){MO.BO=true;}else{MO.BO=false;}: } parc llaa
CUERSP:cadena1 { :MO.BO=!(MO.BO);: } ELS:cadena2 { : MO.BO=true;
if(cadena.contains("true")){RESULT=cadena1;}else{RESULT=cadena2;}: };

```

```

ELS::=elses llaa CUERSP:cadena { :RESULT=cadena;:}

```

```

|{ :RESULT="";:}

```

```

;

```

```

INTE::= INTE:a anddoble INTE:b { :if(a.contains("true") &&
b.contains("true")){RESULT="true";}else{RESULT="false";}: }

```

```

|INTE:a ordoble INTE:b { :if(a.contains("true") ||
b.contains("true")){RESULT="true";}else{RESULT="false";}: }

```

```

|admira INTE:a { :
if(a.contains("true")){RESULT="false";}else{RESULT="true";}: }

```

```

|INTE:a may INTE:b
{ :if(MO.P(a)>MO.P(b)){RESULT="true";}else{RESULT="false";}: }

```

```

|INTE:a men INTE:b
{:if(MO.P(a)<MO.P(b)){RESULT="true";}else{RESULT="false";}:}

|INTE:a menorigual INTE:b
{:if(MO.P(a)<=MO.P(b)){RESULT="true";}else{RESULT="false";}:}

|INTE:a mayorigual INTE:b
{:if(MO.P(a)>=MO.P(b)){RESULT="true";}else{RESULT="false";}:}

|INTE:a diferente INTE:b
{:if(a.toLowerCase().equals(b.toLowerCase())){RESULT="false";}else{RESULT="true";}
:}

|INTE:a igualdoble INTE:b
{:if(a.toLowerCase().equals(b.toLowerCase())){RESULT="true";}else{RESULT="false";}
:}

|para INTE:cadena1 parc { :System.out.println(cadena1);
RESULT="("+cadena1+")";:}

|VAL1:a { : RESULT=a;:}

;

/*

INTE::=INTE:a anddoble INTE:b { :if(a.contains("VER") &&
b.contains("VER")){RESULT="VER";}else{RESULT="FAL";}:}

|INTE:a ordoble INTE:b { :if(a.contains("VER") ||
b.contains("VER")){RESULT="VER";}else{RESULT="FAL";}:}

|admira INTE:a { :
if(a.contains("VER")){RESULT="FAL";}else{RESULT="VER";}:}

|COMPA:a { :if(a.contains("VER")){RESULT="VER";}else{RESULT="FAL";}:}

;*/

COMPA::= COMPA:a may COMPA:b
{:if(MO.P(a)>MO.P(b)){RESULT="VER";}else{RESULT="FAL";};:}

|COMPA:a men COMPA:b
{:if(MO.P(a)<MO.P(b)){RESULT="VER";}else{RESULT="FAL";};:}

|COMPA:a menorigual COMPA:b
{:if(MO.P(a)<=MO.P(b)){RESULT="VER";}else{RESULT="FAL";};:}

|COMPA:a mayorigual COMPA:b
{:if(MO.P(a)>=MO.P(b)){RESULT="VER";}else{RESULT="FAL";};:}

```



```
|COMP A:a diferente COMP A:b  
{:if(a.equals(b)){RESULT="FAL";}else{RESULT="VER";};:}
```

```
|COMP A:a igualdoble COMP A:b  
{:if(a.equals(b)){RESULT="VER";}else{RESULT="FAL";};:}
```

```
|VAL1:a {:RESULT=a;:}
```

```
;/
```

```
/*
```

```
|VAL:a may COMP A:b  
{:if(MO.P(a)>MO.P(b)){RESULT="VER";}else{RESULT="FAL";};:}
```

```
|VAL:a men COMP A:b  
{:if(MO.P(a)<MO.P(b)){RESULT="VER";}else{RESULT="FAL";};:}
```

```
|VAL:a menorigual COMP A:b  
{:if(MO.P(a)<=MO.P(b)){RESULT="VER";}else{RESULT="FAL";};:}
```

```
|VAL:a mayorigual COMP A:b  
{:if(MO.P(a)>=MO.P(b)){RESULT="VER";}else{RESULT="FAL";};:}
```

```
|VAL:a diferente COMP A:b  
{:if(a.equals(b)){RESULT="FAL";}else{RESULT="VER";};:}
```

```
|VAL:a igualdoble COMP A:b  
{:if(a.equals(b)){RESULT="VER";}else{RESULT="FAL";};:}
```

```
*/
```

```
VAL1::= cadena:cadena1 {:RESULT=cadena1;:}
```

```
|booleano:cadena1 {:RESULT=cadena1;:}
```

```
|VALN1:cadena1 {:RESULT=cadena1;:}
```

```
;
```

```
VALN1::=VALN1:cadena1 suma VALN1:cadena2{:RESULT=""+(MO.P(cadena1)+  
MO.P(cadena2));:}
```

```
|VALN1:cadena1 resta VALN1:cadena2{:RESULT=""+(MO.P(cadena1)-  
MO.P(cadena2));:}
```

```
|VALN1:cadena1 multi VALN1:cadena2{:RESULT=""+(MO.P(cadena1)*
MO.P(cadena2));:}
```

```
|VALN1:cadena1 div VALN1:cadena2{:RESULT=""+(MO.P(cadena1)/
MO.P(cadena2));:}
```

```
|VALN1:cadena1 para VALN1:cadena2
parc{:RESULT=""+(MO.P(cadena1)*MO.P(cadena2));:}
```

```
|ANUM:cadena1{:RESULT=cadena1;;}
```

```
|resta VALN1:cadena1{:RESULT="-"+cadena1;;}
```

```
|suma VALN1:cadena1{:RESULT=cadena1;;}
```

```
;
```

```
/*
```

```
//ADORPSKKKadfjopadfjopadfjopadfjopadfjopadfjopadfjopadfjopadfjo
p
```

```
//ADORPSKKKadfjopadfjopadfjopadfjopadfjopadfjopadfjopadfjopadfjo
p
```

```
//ADORPSKKKadfjopadfjopadfjopadfjopadfjopadfjopadfjopadfjopadfjo
p
```

```
//ADORPSKKKadfjopadfjopadfjopadfjopadfjopadfjopadfjopadfjopadfjo
p
```

REVISAR ARRIBA

```
//ADORPSKKKadfjopadfjopadfjopadfjopadfjopadfjopadfjopadfjopadfjo
p
```

```
*/
```

////////PAGINA4

SCRIPT::= interro hs CUERS:cadena interro may {: RESULT= cadena+"\n";:};

CUERS::= VAR:cadena1 CUERS:cadena2 {:RESULT= cadena1+cadena2;;}

```

|ECH:cadena1 CUERS:cadena2 {:RESULT= cadena1+cadena2;;}
|SI:cadena1 CUERS:cadena2 {:RESULT= cadena1+cadena2;;}
|REP:cadena1 CUERS:cadena2 {:RESULT= cadena1+cadena2;;}
|NATIV:cadena1 CUERS:cadena2 {:RESULT= cadena1+cadena2;;}
|TABA:cadena1 CUERS:cadena2 {:RESULT= cadena1+cadena2;;}
|FUBOT:cadena1 CUERS:cadena2 {:RESULT= cadena1+cadena2;;}
| {:RESULT= "";;}
;

```

```

VAR::= dolar id:cadena2 igual VAL:cadena1 puntocomas { : MO.A(cadena2,cadena1,"$");
RESULT="";;};

```

```

ANUM::= entero:cadena1 {:RESULT=cadena1;;}
|decimal:cadena1 {:RESULT=cadena1;;}
|dolar id:cadena1 { : RESULT= MO.R(cadena1,"$");;}
;

```

```

VAL::= cadena:cadena1 {:RESULT=cadena1;;}
|booleano:cadena1 {:RESULT=cadena1;;}
|VALN:cadena1 {:RESULT=cadena1;;}
;

```

```

VALN::=VALN:cadena1 suma VALN:cadena2{:RESULT=""+(MO.P(cadena1)+
MO.P(cadena2));;}

```

```

|VALN:cadena1 resta VALN:cadena2{:RESULT=""+(MO.P(cadena1)-
MO.P(cadena2));;}

```

```

|VALN:cadena1 multi VALN:cadena2{:RESULT=""+(MO.P(cadena1)*
MO.P(cadena2));;}

```

```

|VALN:cadena1 div VALN:cadena2{:RESULT=""+(MO.P(cadena1)/
MO.P(cadena2));;}

```

```

|para VALN:cadena1 parc{:RESULT=""+MO.P("("+cadena1+")");;}

```

```

|VALN:cadena1 para VALN:cadena2
parc{:RESULT=""+(MO.P(cadena1)*MO.P(cadena2));:}

|ANUM:cadena1{:RESULT=""+cadena1;;}

|resta VALN:cadena1{:RESULT=""+"-"+cadena1;;}

|suma VALN:cadena1{:RESULT=""+(cadena1);:}

;

```

```

ECH::=echo CONCAS:cadena1 puntocomas{:int Tamaño=NVeces.size();int Contenido=1;
Contenido=NV; MO.I(cadena1+"\n",Contenido); RESULT="";:};

```

```

CONCAS::=CONCA:cadena1 CONCASS:cadena2 {:RESULT=cadena1+cadena2;;};
CONCASS::= punto CONCAS:cadena1 {:RESULT=cadena1;;}
|{:RESULT="";:}

;

```

```

CONCA::=dolar id:cadena1 {: RESULT=MO.R(cadena1,"$");:}

|cadena:cadena1{:RESULT=cadena1;;}

;

```

////////PAGINA 3

```

IMAP::= path igual IMD:cadena IMAPP:cadena1 {:RESULT="src =
" +(char)34+cadena+(char)34+cadena1;;}

|IMAPP:cadena1 path igual IMD:cadena{:RESULT="src =
" +(char)34+cadena+(char)34+cadena1;;}

|ALT1:cadena path igual IMD:cadena1 ALT2:cadena2{:RESULT="src =
" +(char)34+cadena1+(char)34+cadena+cadena2;;}

|ALT2:cadena path igual IMD:cadena1 ALT1:cadena2{:RESULT="src =
" +(char)34+cadena1+(char)34+cadena+cadena2;;}

|path igual IMD:cadena {:RESULT="src = " +(char)34+cadena+(char)34;;}

```

```

|path igual IMD:cadena ALT1:cadena1 { :RESULT="src =
"+(char)34+cadena+(char)34+cadena1;;}

|path igual IMD:cadena ALT2:cadena1 { :RESULT="src =
"+(char)34+cadena+(char)34+cadena1;;}

;

IMD::= imgdir:cadena1 { :RESULT= cadena1;;}

|cadena:cadena1 { :RESULT= cadena1;;}

;

IMAPP::= ALT1:cadena ALT2:cadena1 { :RESULT= cadena +cadena1;;}

|ALT2:cadena ALT1:cadena1 { :RESULT= cadena +cadena1;;}

;

ALT1::= alto igual entero:cadena { :RESULT=" height= "+(char)34+cadena+(char)34;;}

;

ALT2::= ancho igual entero:cadena { :RESULT=" width= "+(char)34+cadena+(char)34;;}

;

//<button class="btn info">Info</button>

BOT::= boton BOTP:cadena may men div boton may { :RESULT="<button "+cadena+ "
">"+ContenidoBoton+" </button>";};

BOTP::= idboton igual cadena:cadena textoboton igual cadena:cadena1 { :RESULT= "
CLASS= "+cadena; ContenidoBoton=cadena1;;}

|textoboton igual cadena:cadena1 idboton igual cadena:cadena { :RESULT= " CLASS=
"+cadena; ContenidoBoton=cadena1;;}

;

TAB::= tabla borde igual booleano:cadena may men FIL:cadena1 div tabla may
{ :RESULT="\n<TABLE
border="+ (char)34+cadena+(char)34+" ">"+cadena1+"</TABLE>";};

```

FIL::= fila may men COLU:cadena div fila may men
FIL:cadena1 { :RESULT="\n<TR>" + cadena + cadena1;; }

| { :RESULT="";; }

;

COLU::= columnac may MCONN:cadena may men
COLU:cadena2 { :RESULT="\n<TD>" + cadena + cadena2 + "</TD>";; }

| columnac may MCONN:cadena may men

COLU:cadena2 { :RESULT="\n<TD>" + cadena + cadena2 + "</TD>";; }

| { :RESULT="</TR>";; }

;

MCONN::= ESPECIAL:cadena men MCON:cadena1 { :RESULT=" " + cadena + cadena1 + "
";; }

| men MCON:cadena { :RESULT=" " + cadena + " ";; }

;

MCON::= PARRA:cadena MCONN:cadena2 { : RESULT=" " + cadena + cadena2 + " ";; }

| IMA:cadena MCONN:cadena2 { :RESULT=" " + cadena + cadena2 + " ";; }

| BOT:cadena MCONN:cadena2 { :RESULT=" " + cadena + cadena2 + " ";; }

| SCRIPT:cadena MCONN:cadena2 { :RESULT=" " + cadena + cadena2 + " ";; }

| div MCONNN:cadena { :RESULT=cadena;; }

;

MCONNN::= salto may MCONN:cadena { :RESULT="
 " + cadena + " ";; }

| columnac { :RESULT="";; }

| columna { :RESULT="";; }

;

//////////PAGINA 2

SAL::= salto may { :RESULT="
 \n";; } ;

ESPA::= espacio may ESPAP:cadena may { :RESULT=" <DIV> " + cadena + " </DIV>
";};

ESPAP::= CONTE:cadena { :RESULT=cadena;; }

|ESPECIAL:cadena CONTE:cadena1 { :RESULT=cadena+" "+cadena1;; }

;

PARRA::=parrafo ALI:cadena PARRAP:cadena1 men div parrafo may { :RESULT=" <P
"+ cadena +cadena1 +"\n</P> " ;};

PARRAP::=ESPECIAL:cadena { :RESULT=cadena;; }

|men div SAL:cadena { :RESULT=cadena;; }

;

ALI::=alineacion igual ALIP:cadena may { :RESULT="ALIGN= "+ cadena + " >\n";}

|may { :RESULT=">\n";}

;

ALIP::= izquierda { :RESULT=" LEFT ";}

| derecha { :RESULT=" RIGHT ";}

| centrado { :RESULT=" CENTER ";}

| justificado { :RESULT=" JUSTIFY ";}

;

TEXA::=textoa may ESPECIAL:cadena men div textoa may
{ :RESULT="<H1>"+cadena+"</H1>\n";};

TEXB::=textob may ESPECIAL:cadena men div textob may
{ :RESULT="<H3>"+cadena+"</H3>\n";};

IMA::= imagen IMAP:cadena may men div imagen may { :RESULT="<img "+cadena+"
>\n";};

//////// PAGINA1

ESPECIAL::=id:cadena ESPECIAL:cadena1 { :RESULT=cadena+" "+cadena1;; }

```

|entero:cadena ESPECIAL:cadena1 {:RESULT=cadena+" "+cadena1;;}
|decimal:cadena ESPECIAL:cadena1 {:RESULT=cadena+" "+cadena1;;}
|numeral:cadena ESPECIAL:cadena1 {:RESULT=cadena+" "+cadena1;;}
|cadena:val ESPECIAL:cadena1 {:RESULT=val+" "+cadena1;;}
|punto:val ESPECIAL:cadena1 {:RESULT=val+" "+cadena1;;}
|admira:val ESPECIAL:cadena1 {:RESULT=val+" "+cadena1;;}
|cadena:val {:RESULT=val ;;}
|id:cadena {:RESULT=cadena;;}
|entero:cadena {:RESULT=cadena;;}
|decimal:cadena {:RESULT=cadena;;}
|numeral:cadena {:RESULT=cadena;;}
|punto:cadena {:RESULT=cadena;;}
|admira:cadena {:RESULT=cadena;;}

```

;

COMENTA::= men admira ESPECIAL:cadena admira may {:RESULT=cadena;;};

STRUCT::= men compi may CABE:cadena1 CUERP:cadena2 men div compi may
{:setSalida("<HTML>\n"+cadena1+cadena2+"</HTML>");};

CABE::= men cabecera may men titulo may ESPECIAL:cadena men div titulo may men
div cabecera may {:RESULT="<HEAD><TITLE>"+cadena+"</TITLE></HEAD>\n";};

CUERP::=men cuerpo CO:cadena men CONTE:cadena1
may{:RESULT="<BODY>"+cadena+cadena1+"</BODY>\n";};

CONTE::= CONTES:cadena men CONTE:cadena1 {: RESULT=cadena+cadena1;;}

```

|div CONTESS:cadena {: RESULT=cadena;  :}

```


;

CONTESS::= cuerpo{:RESULT="";;}

|espacio{:RESULT="";;}

|salto may men CONTE:cadena {:RESULT="
" + cadena;;}

;

CONTES::= PARRA:cadena {:RESULT=cadena;;}

|SAL:cadena {:RESULT=cadena;;}

|TAB:cadena {:RESULT=cadena;;}

|IMA:cadena {:RESULT=cadena;;}

|TEXA:cadena {:RESULT=cadena;;}

|TEXB:cadena {:RESULT=cadena;;}

|BOT:cadena {:RESULT=cadena;;}

|ESPA:cadena {:RESULT=cadena;;}

|SCRIPT:cadena {:RESULT=cadena;;}

|{:RESULT="";;}

;

//|SAL:cadena {:RESULT=cadena;;}

CO::= fondo igual COL:cadena may {:RESULT=" bgcolor= " + cadena + ">\n ";;}

|may{:RESULT=">\n ";;}

;

```
COL::= color:cadena { :RESULT=cadena;:}
    |rgb:cadena { :RESULT=cadena;:}
;

```

Expresiones Regulares para el analizador Léxico

```
Comentario = "//" [^\r\n]* [^\r\n]
ComentarioMulti = "/*" [^/] ~"*/" | "/*" "/" + "*/"
entero =[0-9]+
decimal =[0-9]+ "." [0-9]*
cadena =["] [^\n]* ["]\n]
letra =[a-zA-ZÑñ]+
la= [\u00E1][\u00E9][\u00ED][\u00F3][\u00FA][\u00F1]
CS=[\u01BA]

id =( {letra} | {la} ) ( {letra} | {entero} | "_" ) *
caracter = "" [ ^ ] ""
esp = \t|f" "\r\n

ComentarioDel = "<!" [^CS] ~"!>" | "<!" "/" + "!>"

```