



Listado de Tokens

TOKEN	PATRON	TIPO
variable	Cualquier carácter inicia en letra o guion bajo seguido de letras o números (reservada)	léxico
letra	Una sola letra mayuscula o minuscula	lexico
entero	Secuencia de dígitos	léxico
llavea	Carácter llave que abre {	léxico
llavec	Carácter llave que cierra }	léxico
puntocomma	Carácter Punto y coma ;	léxico
coma	Coma (,)	léxico
cadena	Inicia en “ o ‘ y finaliza en “ o ‘	léxico
error	Cualquier carácter que no esté en el lenguaje	léxico
comentarios	Inicia en <! y finaliza en !> o solo inicia en //	
mayorque	Carácter >	léxico
ENTERO	Palabra Int(reservada)	léxico
DOUBLE	Palabra double(reservada)	léxico
BOOLEANO	Palabra boolean (reservada)	léxico
CARACTER	Palabra char (reservada)	léxico
VOID	Palabra void (reservada)	léxico
STRING	Palabra string (reservada)	léxico

TRUE	Palabra true (reservada)	léxico
STRING	Palabra false (reservada)	léxico
NEW	Palabra new (reservada)	léxico
INC	Carácter ++	léxico
DEC	Carácter --	léxico
MAS	Carácter +	léxico
MENOS	Carácter -	léxico
MULTIPLICACION	Carácter *	léxico
DIVISION	Carácter /	léxico
POTENCIA	Carácter ^	léxico
MODULO	Carácter %	léxico
IGUALA	Carácter ==	léxico
DIFERENTE	Carácter !=	léxico
MENORIGUALQ	Carácter >=	léxico
MENORQUE	Carácter >	léxico
MAYORIGUALQ	Carácter <=	léxico
MAYORQUE	Carácter <	léxico
IGUAL	Carácter =	léxico
OR	Carácter	léxico

AND	Carácter	léxico
NOT	Caracter j	
MAYORIGUALQ	Carácter >=	léxico
IF	Palabra if (reservada)	léxico
ELSE	Palabra else (reservada)	léxico
SWITCH	Palabra switch (reservada)	léxico
CASE	Palabra case (reservada)	léxico
BREAK	Palabra break (reservada)	léxico
DEFAULT	Palabra default (reservada)	léxico
WHILE	Palabra while (reservada)	léxico
DO	Palabra do (reservada)	léxico
FOR	Palabra for (reservada)	léxico
CONTINUE	Palabra continue (reservada)	léxico
RETURN	Palabra return (reservada)	léxico

Expresiones Regulares

Para la lectura del archivo se utilizaron varias expresiones regulares entre las que se pueden mencionar:

- 1) numero=[0-9]+**
- 2) variables=[a-z|A-Z]+[a-z|A-Z|0-9|"_"]+**
- 3) letrasmin=[a-z] (*letras minúsculas*)**
- 4) comentlinea = ("/*.*\r\n)|("/*.*\n)|("/*.*\r)**
- 5) comentarios="<<!"!""*"([^\>]|[\^!] ">" |'!' "[^\>]) *'"!""*>"**
- 6) cadenas= ["\[^\"]*"["\"]]**
- 7) saltolinea="\n"**
- 8) comillasimplereg="\\\\" (*comillas simples*)**
- 9) comillasdoblereg="\\\\\\" (*comillas dobles*)**
- 10) especial = ("\\n"| "\\\"| "\\'| "\\"'| ""') (*caracteres especiales*)**
- 11) VDOUBLE= [0-9]+(("."[0-9])+)\b**
- 12) VENTERO=[0-9]+\bnúmeros)**
- 13) VCARACTER =**
 (\('((\[\\\][\']|[\\\][\"])|([\\\][n]|[\\\][t]|[\\\][\r]))|(^\\n|'\''{1})))\')

Lo demás fue utilizado con el análisis sintáctico

/*----- 5.10 Precedencia de Operaciones -----*/

Asociación de operadores y precedencia

La precedencia de operadores nos indica la importancia en que una operación debe realizarse por encima del resto. A continuación, se define la misma.

'OR', 'INTERROGA', 'tipo'

'AND'

'NOT'

'IGUALA', 'DIFERENTE', 'MENORQUE', 'MENORIGUALQ', 'MAYORQUE', 'MAYORIGUALQ'

'INC', 'MAS', 'DEC', 'MENOS'

'MULTIPLICACION', 'DIVISION', 'POTENCIA', 'MODULO', 'VARIABLE'

UMENOS

Gramática Independiente del Contexto

COMENTARIOS1 = menorque signoadmira cadena signoadmira mayorque <! cadena- !>
COMENTARIOS2 = slash slash cadena // cadena

Gramática independientes de contexto

/* ===== Definición de la gramatica ===== */

ini : instrucciones EOF

;

instrucciones

: instrucciones instruccion

| instruccion

;

/* ===== INSTRUCCIONES ===== */

instruccion

: declaracion

| funciones

| metodos

| call

| instruccioneswitch

| instruccionif

| instruccionwhile

| instruccionfor

| instrucciondowhile

| instruccionprint

| instruccionprintln

| VARIABLE INC PTCOMA

| VARIABLE DEC PTCOMA

| CONTINUE PTCOMA

| BREAK PTCOMA

| **returns** PTCOMA

| RUN **call**

/*| incredecre { } 5.14 Incremento y Decremento*/

;

/*-----5.17 Sentencias cíclicas-----*/

/*----- 5.19 Funciones -----*/

funciones

: VARIABLE PARIZQ **parametros** PARDER DOSPUNTOS tipo LLAIZQ **instrucciones** LLADER

| VARIABLE PARIZQ PARDER DOSPUNTOS **tipo** LLAIZQ **instrucciones** LLADER

;

/*----- 5.20 Metodos -----*/

metodos

: VARIABLE PARIZQ **parametros** PARDER DOSPUNTOS VOID LLAIZQ **instrucciones** LLADER

| VARIABLE PARIZQ **parametros** PARDER LLAIZQ **instrucciones** LLADER

| VARIABLE PARIZQ PARDER DOSPUNTOS VOID LLAIZQ **instrucciones** LLADER

| VARIABLE PARIZQ PARDER LLAIZQ **instrucciones** LLADER

;

/*----- 5.21 Llamadas -----*/

call

:llamadas PTCOMA

;

llamadas

:VARIABLE PARIZQ paramllamada PARDER

| VARIABLE PARIZQ PARDER

;

paramllamada

:paramllamada COMA expresion

| expresion

;

parametros

:parametros COMA tipo VARIABLE

| tipo VARIABLE

;

/*----- IF -----*/

instruccionif

:IF PARIZQ expresion PARDER LLAIZQ instrucciones LLADER

| IF PARIZQ expresion PARDER LLAIZQ instrucciones LLADER ELSE LLAIZQ instrucciones LLADER

| IF PARIZQ **expresion** PARDER LLAIZQ **instrucciones** LLADER ELSE **instruccionif**

| IF PARIZQ **expresion** PARDER LLAIZQ LLADER

;

/* ----- SWITCH ----- */

instruccionswitch

:SWITCH PARIZQ **expresion** PARDER LLAIZQ **instruccioncaselist instrucciondefault** LLADER

| SWITCH PARIZQ **expresion** PARDER LLAIZQ **instruccioncaselist** LLADER

| SWITCH PARIZQ **expresion** PARDER LLAIZQ **instrucciondefault** LLADER

;

instruccioncaselist

:**instruccioncaselist** CASE **expresion** DOSPUNTOS **instrucciones**

| CASE **expresion** DOSPUNTOS **instrucciones**

;

/* ----- DEFAULT ----- */

instrucciondefault

:DEFAULT DOSPUNTOS **instrucciones** { }

;

/*-----5.17.1. While -----*/

instruccionwhile

```
:WHILE PARIZQ expresion PARDER LLAIZQ instrucciones LLADER  
;
```

/*-----5.17.3. Do-While-----*/

instrucciondowhile

```
:DO LLAIZQ instrucciones LLADER WHILE PARIZQ expresion PARDER PTCOMA  
;
```

/*-----5.17.2. For-----*/

instruccionfor

```
:FOR PARIZQ fordeclarar PTCOMA expresion PTCOMA actualizacion PARDER LLAIZQ  
instrucciones LLADER  
;
```

fordeclarar

```
: tipo VARIABLE IGUAL expresion  
    | VARIABLE IGUAL expresion  
;
```

actualizacion

```
: VARIABLE INC  
    | VARIABLE DEC  
    | VARIABLE IGUAL expresion  
;
```

/*----- 5.22 Función Print -----
*/

instruccionprint

:PRINT PARIZQ **expresion** PARDER PTCOMA

;

/*----- 5.23 Función Println -----*/

instruccionprintln

:PRINTLN PARIZQ **expresion** PARDER PTCOMA

;

/*----- RUN -----*/

instruccionrun

: RUN VARIABLE PARA PARC

| RUN VARIABLE PARA **listavalores** PARC

;

/*----- RETURN -----*/

returns

: RETURN **expresion**

| RETURN

;

/*----- DECLARACION ASIGNACION VARIABLES -----*/

declaracion

: **tipo** VARIABLE PTCOMA

| **tipo** VARIABLE IGUAL **expresion** PTCOMA

```

| tipo notacioncomas PTCOMA

| tipo notacioncomas IGUAL expresion PTCOMA

| tipo VARIABLE CORIZQ CORDER IGUAL NEW tipo CORIZQ expresion CORDER PTCOMA

| tipo VARIABLE CORIZQ CORDER IGUAL NEW tipo CORIZQ expresion CORDER CORIZQ
expresion CORDER PTCOMA

| tipo VARIABLE CORIZQ CORDER IGUAL CORIZQ listavalores CORDER PTCOMA

| tipo VARIABLE IGUAL VARIABLE CORIZQ expresion CORDER PTCOMA

| tipo VARIABLE IGUAL VARIABLE CORIZQ expresion CORDER CORIZQ expresion CORDER
PTCOMA

| notacioncomas IGUAL expresion PTCOMA

| VARIABLE IGUAL expresion PTCOMA

```

/*5.15.1.3 Modificación de Vectores*/

```

| VARIABLE CORIZQ expresion CORDER IGUAL expresion PTCOMA

| VARIABLE CORIZQ expresion CORDER CORIZQ expresion CORDER IGUAL expresion
PTCOMA

;

```

notacioncomas

```

: notacioncomas COMA VARIABLE

| VARIABLE

;

```

/* ----- EXPRESIONES ----- */

expresion

: MENOS expresion

| NOT expresion

| expresion MAS expresion

| expresion MENOS expresion

| expresion MULTIPLICACION expresion

| expresion DIVISION expresion

| expresion MODULO expresion

| expresion POTENCIA expresion

| expresion MENORIGUALQ expresion

| expresion MENORQUE expresion

| expresion MAYORIGUALQ expresion

| expresion MAYORQUE expresion

| expresion IGUALA expresion

| expresion DIFERENTE expresion

| expresion OR expresion

| expresion AND expresion

| PARA tipo PARC expresion

| VARIABLE

| VENTERO

| VDOUBLE

| CADENA

| VCHARACTER

| TRUE

| FALSE

| PARIZQ **expresion** PARDER

| VARIABLE CORIZQ **expresion** CORDER

| VARIABLE CORIZQ **expresion** CORDER CORIZQ **expresion** CORDER

| **expresion** INTERROGA **expresion** DOSPUNTOS **expresion**

| TOLOWER PARIZQ **expresion** PARDER

| TOUPPER PARIZQ **expresion** PARDER

| ROUND PARIZQ DOUBLE PARDER

| TYPEOF PARIZQ **expresion** PARDER

| TOSTRING PARIZQ **expresion** PARDER

| LENGTH PARIZQ **expresion** PARDER

| TYPEOF PARIZQ **expresion** PARDER

| TOSTRING PARIZQ **expresion** PARDER

| TOCHARARRAY PARIZQ **expresion** PARDER

| **llamadas**

;

listavalores

: **listavalores** COMA **expresion**

| **expresion**

;

/* ----- TIPOS ----- */

tipo

:ENTERO

|DOUBLE

|BOOLEANO

|CHARACTER

|STRING

|VOID

;