

```
// Terminals

terminal VAR, INT, DOUBLE, BOOL, CHAR, STRINGTYPE;
terminal IF, ELSE;
terminal WHILE, FOR, DO;
terminal BREAK, CONTINUE;
terminal PRINT;
terminal String CADENA, ENTERO, DECIMAL;
terminal char CARACTER;
terminal boolean BOLEANO;
terminal String IDENTIFICADOR;

terminal MAS, MENOS, MULT, DIV, POTENCIA, MODULO;
terminal INCREMENTO, DECREMENTO;
terminal IGUAL, DIFERENTE, MENOR, MENOR_IGUAL, MAYOR,
MAYOR_IGUAL;
terminal AND, OR, XOR, NOT;
terminal PAR_ABRIR, PAR_CERRAR;
terminal LLAVE_ABRIR, LLAVE_CERRAR;
terminal PUNTO_COMA, DOS_PUNTOS;
terminal SWITCH, CASE, DEFAULT;
terminal ASIG;
terminal UMINUS;
terminal COMA, START, VOID;
terminal ROUND, TOSTRING;
terminal RETURN;
terminal COR_ABRIR, COR_CERRAR;
terminal APPEND, REMOVE;
terminal LIST, NEW;
terminal PUNTO;
terminal LENGTH, FIND;
```

// Nonterminals

nonterminal LinkedList INICIO, INSTRUCCIONES, LISTA_CASES,
ELSEIF_LIST, DEFAULT_INST;
nonterminal Instrucion INSTRUCCION, EXPRESION, DECLARACION,
ASIGNACION, SIF, WHILE_LOOP, DO WHILE_LOOP, FOR_LOOP,
BREAK_INST, CONTINUE_INST, SWITCH_INST;
nonterminal SwitchCaso CASE_INST;
nonterminal Tipo TIPO;
nonterminal LinkedList<HashMap> PARAMS;
nonterminal Instrucion STARTT, LLAMADA;
nonterminal LinkedList<Instrucion> PARAMSLLAMADA;
nonterminal LinkedList<Instrucion> LISTA_EXPR;
nonterminal Instrucion FUNCION_DECL;
nonterminal Instrucion DECLARACIONLISTA;
nonterminal Tipo TIPO_BASE;

// Precedencia

precedence left OR;
precedence left AND;
precedence left XOR;
precedence right NOT;
precedence left IGUAL, DIFERENTE, MENOR, MAYOR, MENOR_IGUAL,
MAYOR_IGUAL;
precedence left MAS, MENOS;
precedence left MULT, DIV, MODULO;
precedence right POTENCIA;
precedence right UMINUS;
precedence left COR_ABRIR;

start with INICIO;

// Reglas de la gramática

INICIO ::= INSTRUCCIONES;

INSTRUCCIONES ::= INSTRUCCIONES INSTRUCCION
| INSTRUCCION;

INSTRUCCION ::= PRINT PAR_ABRIR EXPRESION PAR_CERRAR
PUNTO_COMA

| DECLARACION PUNTO_COMA
| DECLARACION
| ASIGNACION PUNTO_COMA
| ASIGNACION
| SIF
| WHILE_LOOP
| DO_WHILE_LOOP
| FOR_LOOP
| SWITCH_INST
| BREAK_INST
| CONTINUE_INST
| IDENTIFICADOR INCREMENTO
| IDENTIFICADOR DECREMENTO
| IDENTIFICADOR INCREMENTO PUNTO_COMA
| IDENTIFICADOR DECREMENTO PUNTO_COMA
| STARTT
| RETURN EXPRESION PUNTO_COMA
| RETURN PUNTO_COMA
| IDENTIFICADOR COR_ABRIR EXPRESION COR_CERRAR ASIG
EXPRESION PUNTO_COMA
| IDENTIFICADOR COR_ABRIR EXPRESION COR_CERRAR
COR_ABRIR EXPRESION COR_CERRAR ASIG EXPRESION PUNTO_COMA
| FUNCION_DECL
| LLAMADA

| IDENTIFICADOR PUNTO APPEND PAR_ABRIR EXPRESION
PAR_CERRAR PUNTO_COMA
| IDENTIFICADOR PUNTO REMOVE PAR_ABRIR EXPRESION
PAR_CERRAR PUNTO_COMA
| DECLARACIONLISTA
| DECLARACIONLISTA PUNTO_COMA;

WHILE_LOOP ::= WHILE PAR_ABRIR EXPRESION PAR_CERRAR
LLAVE_ABRIR INSTRUCCIONES LLAVE_CERRAR;

DO WHILE_LOOP ::= DO LLAVE_ABRIR INSTRUCCIONES
LLAVE_CERRAR WHILE PAR_ABRIR EXPRESION PAR_CERRAR
PUNTO_COMA;

FOR_LOOP ::=

FOR PAR_ABRIR DECLARACION PUNTO_COMA EXPRESION
PUNTO_COMA INSTRUCCION PAR_CERRAR
LLAVE_ABRIR INSTRUCCIONES LLAVE_CERRAR
| FOR PAR_ABRIR ASIGNACION PUNTO_COMA EXPRESION
PUNTO_COMA INSTRUCCION PAR_CERRAR
LLAVE_ABRIR INSTRUCCIONES LLAVE_CERRAR;

SIF ::= IF PAR_ABRIR EXPRESION PAR_CERRAR LLAVE_ABRIR
INSTRUCCIONES LLAVE_CERRAR

| IF PAR_ABRIR EXPRESION PAR_CERRAR LLAVE_ABRIR
INSTRUCCIONES LLAVE_CERRAR ELSE LLAVE_ABRIR INSTRUCCIONES
LLAVE_CERRAR
| IF PAR_ABRIR EXPRESION PAR_CERRAR LLAVE_ABRIR
INSTRUCCIONES LLAVE_CERRAR ELSEIF_LIST
| IF PAR_ABRIR EXPRESION PAR_CERRAR LLAVE_ABRIR
INSTRUCCIONES LLAVE_CERRAR ELSEIF_LIST ELSE LLAVE_ABRIR
INSTRUCCIONES LLAVE_CERRAR;

ELSEIF_LIST ::= ELSEIF_LIST ELSE IF PAR_ABRIR EXPRESION
PAR_CERRAR LLAVE_ABRIR INSTRUCCIONES LLAVE_CERRAR

| ELSE IF PAR_ABRIR EXPRESION PAR_CERRAR LLAVE_ABRIR
INSTRUCCIONES LLAVE_CERRAR;

BREAK_INST ::= BREAK PUNTO_COMA;

CONTINUE_INST ::= CONTINUE PUNTO_COMA;

DECLARACION ::=

 VAR IDENTIFICADOR DOS_PUNTOS TIPO ASIG EXPRESION
 | VAR IDENTIFICADOR DOS_PUNTOS TIPO;

LISTA_CASES ::= LISTA_CASES CASE_INST
 | CASE_INST;

CASE_INST ::= CASE EXPRESION DOS_PUNTOS INSTRUCCIONES;

DEFAULT_INST ::= DEFAULT DOS_PUNTOS INSTRUCCIONES
 | ;

SWITCH_INST ::= SWITCH PAR_ABRIR EXPRESION PAR_CERRAR
LLAVE_ABRIR LISTA_CASES DEFAULT_INST LLAVE_CERRAR
 | SWITCH PAR_ABRIR EXPRESION PAR_CERRAR LLAVE_ABRIR
DEFAULT_INST LLAVE_CERRAR;

ASIGNACION ::= IDENTIFICADOR ASIG EXPRESION;

PARAMS ::= PARAMS COMA TIPO_BASE IDENTIFICADOR
 | TIPO_BASE IDENTIFICADOR;

STARTT ::= START IDENTIFICADOR PAR_ABRIR PAR_CERRAR
PUNTO_COMA

| START IDENTIFICADOR PAR_ABRIR PARAMSLLAMADA
PAR_CERRAR PUNTO_COMA;

PARAMSLLAMADA ::= PARAMSLLAMADA COMA EXPRESION

| EXPRESION;

LLAMADA ::= IDENTIFICADOR PAR_ABRIR PARAMSLLAMADA
PAR_CERRAR PUNTO_COMA

| IDENTIFICADOR PAR_ABRIR PAR_CERRAR PUNTO_COMA;

LISTA_EXPR ::=

LISTA_EXPR COMA EXPRESION

| EXPRESION

| ;

FUNCION_DECL ::=

TIPO IDENTIFICADOR PAR_ABRIR PAR_CERRAR LLAVE_ABRIR
INSTRUCCIONES LLAVE_CERRAR

| TIPO IDENTIFICADOR PAR_ABRIR PARAMS PAR_CERRAR
LLAVE_ABRIR INSTRUCCIONES LLAVE_CERRAR;

DECLARACIONLISTA ::=

LIST MENOR TIPO_BASE MAYOR IDENTIFICADOR ASIG NEW LIST
PAR_ABRIR PAR_CERRAR;

TIPO ::= TIPO_BASE

| TIPO_BASE COR_ABRIR COR_CERRAR

| TIPO_BASE COR_ABRIR COR_CERRAR COR_ABRIR COR_CERRAR;

TIPO_BASE ::= INT

| DOUBLE

| BOOL

```
| CHAR  
| STRINGTYPE  
| VOID;
```

// --- EXPRESSIONS ---

EXPRESION ::=

```
// Operadores lógicos  
EXPRESION OR EXPRESION  
| EXPRESION AND EXPRESION  
| EXPRESION XOR EXPRESION  
| NOT EXPRESION
```

```
// Relacionales  
| EXPRESION IGUAL EXPRESION  
| EXPRESION DIFERENTE EXPRESION  
| EXPRESION MENOR EXPRESION  
| EXPRESION MAYOR EXPRESION  
| EXPRESION MENOR_IGUAL EXPRESION  
| EXPRESION MAYOR_IGUAL EXPRESION
```

```
// Aritméticos  
| EXPRESION MAS EXPRESION  
| EXPRESION MENOS EXPRESION  
| EXPRESION MULT EXPRESION  
| EXPRESION DIV EXPRESION  
| EXPRESION MODULO EXPRESION  
| EXPRESION POTENCIA EXPRESION  
| MENOS EXPRESION %prec UMINUS
```

// Casting y paréntesis

```
| PAR_ABRIR TIPO PAR_CERRAR EXPRESION
| PAR_ABRIR EXPRESION PAR_CERRAR

// Literales
| ENTERO
| DECIMAL
| CARACTER
| CADENA
| BOOLEANO

// Identificador simple
| IDENTIFICADOR

// Funciones nativas
| ROUND PAR_ABRIR EXPRESION PAR_CERRAR
| TOSTRING PAR_ABRIR EXPRESION PAR_CERRAR
| LENGTH PAR_ABRIR EXPRESION PAR_CERRAR
| IDENTIFICADOR PUNTO FIND PAR_ABRIR EXPRESION PAR_CERRAR
| IDENTIFICADOR PUNTO APPEND PAR_ABRIR EXPRESION
PAR_CERRAR
| IDENTIFICADOR PUNTO REMOVE PAR_ABRIR EXPRESION
PAR_CERRAR

// Acceso a matrices y vectores
| IDENTIFICADOR COR_ABRIR EXPRESION COR_CERRAR COR_ABRIR
EXPRESION COR_CERRAR
| IDENTIFICADOR COR_ABRIR EXPRESION COR_CERRAR

// Inicializador de vector
| COR_ABRIR LISTA_EXPR COR_CERRAR

// Llamadas a función
```

| IDENTIFICADOR PAR_ABRIR PAR_CERRAR
| IDENTIFICADOR PAR_ABRIR PARAMS LLAMADA PAR_CERRAR;