

=====

## GRAMÁTICA DEL LENGUAJE – PROYECTO OLC1

=====

### PROGRAMA

-----

programa

→ items

items

→ items item

| item

item

→ instruccion

| metodo

| funcion

| start

### START

-----

start

→ START ID '(' ')' ';' ;

### INSTRUCCIONES

-----

instruccion

→ PRINT '(' expresion ')' ';' ;

| VAR ID ':' tipo ';' ;

```
| VAR ID ':' tipo '=' expresion ';'
| VAR ID ':' tipo dimensiones ';'
| VAR ID ':' tipo dimensiones '=' expresion ';'
| ID '=' expresion ';'
| ID indices '=' expresion ';'
| ID '++' ';'
| ID '--' ';'
| llamada ';'
| bloque
| IF '(' expresion ')' bloque
| IF '(' expresion ')' bloque ELSE bloque
| WHILE '(' expresion ')' bloque
| DO bloque WHILE '(' expresion ')' ';'
| FOR '(' forInit ';' expresion ';' forUpdate ')' bloque
| BREAK ';'
| CONTINUE ';'
| RETURN expresion ';'
| RETURN ';'
| SWITCH '(' expresion ')' '{' casos '}'
```

## BLOQUES

-----

bloque

→ '{' '}'

| '{' items '}'

## TIPOS

.....

tipo

→ int  
| double  
| bool  
| char  
| string

## VECTORES Y MATRICES

-----

dimensiones

→ dimensiones '[' '']  
| '[' '']

indices

→ indices '[' expresion '']  
| '[' expresion '']

listaExp

→ listaExp ',' expresion  
| expresion

## FUNCIONES Y MÉTODOS

-----

metodo

→ void ID '(' ')' bloque  
| void ID '(' parametros ')' bloque

funcion

→ tipo ID '(' ')' bloque  
| tipo ID '(' parametros ')' bloque

parametros

→ parametros ',' parametro  
| parametro

parametro

→ tipo ID

LLAMADAS

-----

llamada

→ ID '(' '  
| ID '(' argumentos ')'

argumentos

→ argumentos ',' expresion  
| expresion

SWITCH

-----

casos

→ casos caso  
| caso

caso

→ CASE expresion ':' items  
| DEFAULT ':' items

FOR

---

forInit

→ ID '=' expresion

forUpdate

→ ID '=' expresion

| ID '++'

| ID '--'

EXPRESIONES

-----

expresion

→ expresion '+' expresion

| expresion '-' expresion

| expresion '\*' expresion

| expresion '/' expresion

| expresion '%' expresion

| expresion '^' expresion

| expresion '<' expresion

| expresion '<=' expresion

| expresion '>' expresion

| expresion '>=' expresion

| expresion '==' expresion

| expresion '!=' expresion

| expresion '&&' expresion

| expresion '||' expresion

| expresion '^' expresion

| '!' expresion

| '-' expresion

| '(' expresion ')'

| literal

| acceso

| llamada

| vectorLiteral

## LITERALES

-----

literal

→ INT\_LIT

| DOUBLE\_LIT

| STRING\_LIT

| CHAR\_LIT

| true

| false

## ACCESOS

-----

acceso

→ ID

| ID indices

## VECTOR LITERAL

-----

vectorLiteral

→ '[' '']'

| '[' listaExp '']'

## FUNCIONES NATIVAS

-----

nativa

→ length '(' expresion ')'  
| round '(' expresion ')'  
| toString '(' expresion ')'  
| ID '.' find '(' expresion ')'  
| ID '.' append '(' expresion ')'  
| ID '.' remove '(' expresion ')'

=====

FIN DE LA GRAMÁTICA

=====