### **TERMINALES**

### **No TERMINALES**

INSTRUCCIONES

**INSTRUCCION** 

**BLOQUE\_SENTENCIAS** 

ASIGNACION

DECLARACION

**MODIFICADOR** 

LLAMADA\_FUNCION
DECLARACION\_METODOS
NO IMPLEMENTADAS

**EXEC** 

SENTENCIA\_CICLICA
SENTENCIA\_CONTROL

EXPRESION UNMENOS DATO

TIPO\_DATO

IF ELSE ELIF

CONTROL\_SWITCH

SWITCH CASELIS CASES CASE DEFAULTS DEFAULT

SENTENCIA\_CICLICA
SENTENCIA\_WHILE

WHILE

SENTENCIA\_FOR ENCABEZADOFOR

**FOR** 

SENTENCIA\_DOWHILE

DO

LLAMADA\_FUNCION LLAMADA\_METODO INICIOMETODO

**PARAMETOS LLAMADA** 

FUNCION\_PRINT

**DECLARACION METODOS** 

METODO VOID

**PARAMETROS** 

**NO\_IMPLEMENTADAS** 

func\_exec

def\_entero

def\_decimal

def\_caracter

def cadena

def\_boolean

esc salto linea

esc\_saito\_iiilea

esc\_comilla\_doble

esc\_tabulacion

esc\_comilla\_simple

esc\_barra\_invertida

incremento

decremento

suma

resta

multiplicacion

division

potencia

op modulo

op doble igual

op menor igual

op\_mayor\_igual

op\_diferencia

op\_igual

op\_menor

op\_mayor

dos\_puntos

pregunta\_cierra

op\_or op and

op\_and

op\_not

par\_abre par\_cierra

punto\_coma

llave\_abre

llave\_cierra

coma

continue

| if

else

switch

case

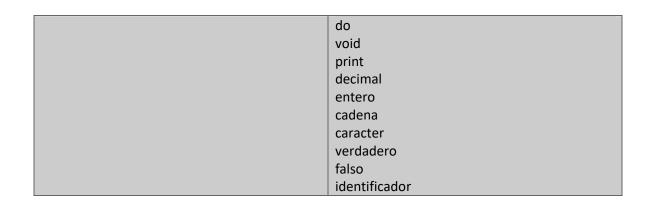
break

default

. ..

while

for



## **Expresiones Regulares**

```
[0-9]+("."[0-9]+)\b r
[0-9]+\b
("\"")([^\"\\]|\\.)*("\"")
```

### Inicio de Gramática

# **GRAMÁTICA**

```
BLOQUE_SENTENCIAS: llave_abre INSTRUCCIONES llave_cierra
```

| llave\_abre llave\_cierra

error llave\_cierra

INSTRUCCION: ASIGNACION punto\_coma

|DECLARACION punto\_coma

|MODIFICADOR punto\_coma

|LLAMADA\_FUNCION punto\_coma

|DECLARACION\_METODOS

|NO\_IMPLEMENTADAS

|EXEC punto\_coma

|SENTENCIA\_CICLICA

|SENTENCIA\_CONTROL

DECLARACION: TIPO\_DATO identificador

ASIGNACION: identificador op\_igual EXPRESION

|TIPO\_DATO identificador op\_igual EXPRESION

EXPRESION: resta EXPRESION

|par\_abre TIPO\_DATO par\_cierra EXPRESION

**EXPRESION** suma EXPRESION

**EXPRESION** resta EXPRESION

**EXPRESION** multiplicacion EXPRESION

**EXPRESION division EXPRESION** 

EXPRESION potencia EXPRESION

|EXPRESION op\_modulo EXPRESION

|EXPRESION op\_and EXPRESION

|EXPRESION op\_or EXPRESION

|op\_not EXPRESION

|EXPRESION op\_mayor EXPRESION

```
|EXPRESION op_menor EXPRESION
      |EXPRESION op_mayor_igual EXPRESION{$$
      |EXPRESION op_menor_igual EXPRESION
      |EXPRESION op_doble_igual EXPRESION
      |EXPRESION op_diferencia EXPRESION
      |EXPRESION pregunta_cierra EXPRESION dos_puntos EXPRESION
      EXPRESION incremento
      EXPRESION decremento
      DATO
DATO: decimal
    entero
    verdadero
    |falso
    cadena
    caracter
    |identificador
SENTENCIA_CONTROL: IF BLOQUE_SENTENCIAS else IF BLOQUE_SENTENCIAS ELIF
        | IF BLOQUE_SENTENCIAS
        | IF BLOQUE_SENTENCIAS ELSE BLOQUE_SENTENCIAS
        |CONTROL_SWITCH
IF: if par_abre EXPRESION par_cierra
ELSE: else
ELIF: else IF BLOQUE_SENTENCIAS ELIF
    | ELSE BLOQUE_SENTENCIAS
CONTROL_SWITCH: switch SWITCH llave_abre CASELIST llave_cierra
```

SWITCH: par abre EXPRESION par cierra

**CASELIST: CASES CASELIST** 

**ICASES** 

**CASES: CASE INSTRUCCIONES** 

|CASE

|DEFAULTS;

CASE: case EXPRESION dos\_puntos

**DEFAULTS: DEFAULT INSTRUCCIONES** 

DEFAULT: default dos\_puntos

SENTENCIA\_CICLICA: SENTENCIA\_WHILE

| SENTENCIA\_FOR

|SENTENCIA\_DOWHILE

SENTENCIA\_WHILE: WHILE BLOQUE\_SENTENCIAS

WHILE: while par\_abre EXPRESION par\_cierra

SENTENCIA\_FOR: ENCABEZADOFOR BLOQUE\_SENTENCIAS

ENCABEZADOFOR: FOR ASIGNACION punto\_coma EXPRESION punto\_coma MODIFICADOR

par\_cierra

FOR ASIGNACION punto\_coma EXPRESION punto\_coma ASIGNACION par\_cierra

FOR: for par\_abre

SENTENCIA\_DOWHILE: DO BLOQUE\_SENTENCIAS while par\_abre EXPRESION par\_cierra

punto\_coma

DO:do

LLAMADA\_FUNCION: FUNCION\_PRINT

|LLAMADA\_METODO

LLAMADA\_METODO: INICIOMETODO PARAMETOS\_LLAMADA par\_cierra

| INICIOMETODO par\_cierra

INICIOMETODO: identificador par\_abre

PARAMETOS\_LLAMADA: EXPRESION coma PARAMETOS\_LLAMADA

**EXPRESION** 

FUNCION\_PRINT: print par\_abre EXPRESION par\_cierra

DECLARACION\_METODOS: METODO BLOQUE\_SENTENCIAS

METODO: VOID identificador par\_abre par\_cierra

|VOID identificador par\_abre PARAMETROS par\_cierra

VOID: void

PARAMETROS: PARAMETRO coma PARAMETROS

| PARAMETRO;

PARAMETRO: TIPO\_DATO identificador

EXEC: func\_exec LLAMADA\_METODO;

NO\_IMPLEMENTADAS: continue;