Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Organización de Lenguajes y compiladores 1 – “C”
Laboratorio
Aux. José Diego Pérez Toralla
Aux. Mynor

Proyecto 2

Gramática

Nombre: Eduardo Alexander Reyes Gonzalez

Carnet: 202010904

Expresiones Regulares

digit	[0-9]
corchete_abre	"["
corchete_cierra	"]"
doblediagonal	"\\"
int	(?:[0-9]|[1-9][0-9]+)
EXPRESION	(?:[eE][-+]?[0-9]+)
frac	(?:\.[0-9]+)

Las expresiones Regulares utilizadas en la gramática son las siguientes:

- ID: ([a-zA-ZÑñ]|("_"[a-zA-ZÑñ]))([a-zA-ZÑñ]|[0-9]|"_")*
- CADENA: \"(?:[{corchete_abre}|{corchete_cierra}]|["\\"]["bnrt/["\\"]]|[^"["\\"])*\"
- CARÁCTER: \'((\\\')|(\\\")|(\\\\)|(\\"n")|(\\"t")|[^\\\n\'])?\'
- DECIMALES {int}{frac}\b
- ENTERO: {int}\b
- ESPACIO EN BLANCO: \s+
- FINAL DE CADENA: <<EOF>>
- COMENTARIO SIMPLE: "//".*
- COMENTARIO MULTILINEA: [/][*][^*]*[*]+([^/*][^*]*[*]+)*[/]

Terminales

Los terminales representan elementos concretos que aparecen en la cadena final de la gramática, no pueden descomponerse en partes mas pequeñas y no pueden ser reemplazadas

La lista de terminales utilizados en la gramática es la siguiente:

TERMINAL	CUANDO RECONOCE
RINT'	→ "int"
RDOUBLE	→ "double"
RBOOLEAN	→ “boolean“
RCHAR'	→ “char”
RSTRING	→ “string”
RTRUE	→ “true”
RFALSE	→ “false”
RIF	→ “if”
RPRINT	→”print”
RELSE	→ “else”
RVOID	→”void”
RRETURN	→ “return”
RSWITCH	→ “switch”
RCASE	→ “case”
RDEFAULT	→”default”
RWHILE	→ “while”
RFOR	→ “for”
RDO	→ “do”
RBREAK	→ “break”
RCONTINUE	→ “continue”
RRETURN	→ “return”
RNEW	→ “new”
RLIST	→ “list”
RADD	→ “add”
RTOLOWER	→ “tolower”
RTOUPPER	→ “toupper”
RLENGTH	→ “length”
RTRUNCATE	→ “truncate”
RROUND	→ “round”
RTYPEOF	→ “typeof”
RTOSTRING	→ “tostring”
RTOCHARARRAY	→ “tochararray”
RMAIN	→ “main”

Los terminales que representan símbolos, fueron representados con el mismo signo.

La lista de terminales que representan signos utilizados en la gramática es la siguiente:

"$"
"++"
"--"
"+"
"-"
"*"
"/"
"^"
"%"
"("
")"
"=="
"="
" "
","
"."
";"
"||"
"&&"
"!="
"!"
"<="
">="
">"
"<"
"{"
"}"
"["
"]"
"?"
":"
"\n"
"\'"
"\\\\"
"\""
"\t"

No Terminales

Los no terminales son símbolos que representan una categoría gramatical o una estructura sintáctica. Estos no aparecen en la cadena final de la gramática, sino que se utilizan para generarla.

La lista de No terminales utilizada en la gramática es la siguiente:

Ini	→	Indica el inicio de la lectura
Instrucciones	→	Obtiene las instrucciones
Instrucción	→	Obtiene una instrucción
FUNCION_MAIN	→	reconoce la instrucción main
DECLARACION_VARIABLE	→	representa una declaración de variable
DECLARACION_VECTORES	→	representa una declaración de vector
DECLARACION_LISTAS	→	representa una declaración de lista
AGREGAR_DATO_LISTAS	→	representa una inserción a lista
LISTA_EXPRESIONES	→	representa lista de expresión en funciones
DECLARACION_METODO	→	representa una declaración de métodos
INSTRUCCIONES_FUNCION	→	representa la lista de ejecuciones de funciones void.
DECLARACION_FUNCION	→	representa una declaración de funciones
PARAMETROS	→	representa los parámetros que se inicializan en las funciones y métodos
DECLARACION_VACIA_METODO	→	representa una declaración de variable utilizado por los parámetros
LLAMADA_METODOS	→	representa una llamada de métodos

PARAMETROS_LLAMADA	→	representa los parámetros que se envían a los métodos.
ASIGNACION_VARIABLE	→	representa una asignación de valor a variables
ASIGNACION_VECTORES	→	representa una asignación de valor a vectores
ASIGNACION_LISTAS	→	representa una asignación de valor a listas
INSTRUCCIONES_FUNCION	→	representa las instrucciones de una función
FUNCION_IF	→	representa la declaración de una función if
FUNCION_PRINT	→	representa la declaración de una función print
FUNCION_WHILE	→	representa la declaración de una función while
FUNCION_FOR	→	representa la declaración de una función for
FUNCION_DOWHILE	→	representa la declaración de una función do-while
DEC_O_ASIG	→	representa la declaración o asignación de variable para la función for
FUNCION_SWITCH	→	representa la declaración de una función switch
CASES_SWITCH	→	representa la declaración de un case dentro de un switch
CASE_SWITCH	→	representa una declaración de un case
DEFAULT	→	representa la declaración de un default para la función switch

INSTRUCCIONES_SWITCH	→	representa las instrucciones que se utilizan dentro de un case o default
SENTENCIA_BREAK	→	representa una declaración de una función break
SENTENCIA_CONTINUE	→	representa una declaración de una función continue
SENTENCIA_RETURN	→	representa una declaración de una función return utilizada para retornar valores o salir de funciones
TIPO	→	representa un tipo de dato primitivo
EXPRESION	→	representa una expresión que puede ser: una operación aritmética, números, comparaciones, entre otros.

GRAMATICA

```
%lex
%options case-insensitive
digit                       [0-9]
corchete_abre               "["
corchete_cierra             "]
doblediagonal               "\\"
int                         (?:[0-9]|[1-9][0-9]+)
EXPRESION                         (?:[eE][-+]?[0-9]+)
frac                        (?:\.[0-9]+)
%%
\s+
<<EOF>>
/* COMENTARIO SIMPLE Y MULTILINEA */
"//".*
[/][*][^*]*[*]+([^/*][^*]*[*]+)*[/]


/* ---------------------- PALABRAS RESERVADAS --------------------*/

"int"
"double"
"boolean"
"char"
"string"
"true"
"false"
"if"
"print"
"else"
"void"
"return"
"switch"
"case"
"default"
"while"
"for"
"do"
"break"
"continue"
"return"
"new"
"list"
"add"
"toLower"
"toUpper"
```

```
"length"
"truncate"
"round"
"typeof"
"toString"
"toCharArray"
"main"

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

([a-zA-ZÑñ]|("_"[a-zA-ZÑñ]))([a-zA-ZÑñ]|[0-
9]|"_")*
\"(?:[{corchete_abre}|{corchete_cierra}]|["\\"]["bnrt/["\\"]]|[^"["\\"])*
\"

\'((\\\')|(\\\")|(\\\\)|(\\"n")|(\\"t")|[^\\\n\'])?\'

{int}{frac}\b

{int}\b


/*--------------------------SIGNOS-------------------------------*/

"$"
"++"
"--"
"+"
"-"
"*"
"/"
"^"
"%"
"("
")"
"=="
"="
"¸"
":"
";"
"||"
"&&"
"!="
"!"
"<="
">="
">"
```

```
"<"
"{"
"}"
"["
"]"
"?"
"."
"\n"
"\'"
"\\\\"
"\""
"\t"

.
/lex
/* Asociacion de operadores y precedencia */
/*Operaciones logicas*/
%left '?' ':'
%left '++' '--'
%left '||'
%left '&&'
%left '!'
%left '==' '!=' '<' '<=' '>' '>='
%left '+' '-'
%left '*' '/' '%'
%left '^'
%right negativo '!' '('

%start ini

%%
```

```
ini    : instrucciones EOF
;

instrucciones :      instrucciones
                          | instruccion
;
instruccion :                 DECLARACION_VARIABLE ';'
                            |   DECLARACION_VECTORES ';'
                            |   DECLARACION_LISTAS ';'
                            |   DECLARACION_METODO
                            |   DECLARACION_FUNCION
                            |   LLAMADA_METODOS
                            |   ASIGNACION_VARIABLE  ';'
                            |   ASIGNACION_VECTORES
                            |   ASIGNACION_LISTAS
                            |   AGREGAR_DATO_LISTAS  ';'
                            |   FUNCION_IF
                            |   FUNCION_PRINT ';'
                            |   FUNCION_WHILE
                            |   FUNCION_FOR
                            |   FUNCION_DO_WHILE ';'
                            |   FUNCION_SWITCH
                            |   FUNCION_TO_LOWER ';'
                            |   FUNCION_MAIN ';'
                            |   SENTENCIA_BREAK
                            |   SENTENCIA_CONTINUE
                            |   SENTENCIA_RETURN
                            |   error PTCOMA
                            |   error
;

FUNCION_MAIN:               RMAIN id '('PARAMETROS_LLAMADA')'
                           |RMAIN id ('')'
;

DECLARACION_VARIABLE :        TIPO id ‘=' EXPRESION
                            | TIPO id
;
```

```
DECLARACION_VECTORES:  TIPO '[' ']' id '=' 'RNEW' TIPO ['EXPRESION']'
                     | TIPO '[' ']' id '=' '{' LISTA_EXPRESIONES '}'
;


DECLARACION_LISTAS: RLIST '<' TIPO '>' id '=' RNEW RLIST '<' TIPO '>'
;


AGREGAR_DATO_LISTAS:  id '.' RADD '(' EXPRESION ')'
;


LISTA_EXPRESIONES:    LISTA_EXPRESIONES ',' EXPRESION
                    | EXPRESION
;


DECLARACION_METODO:   RVOID id '('PARAMETROS')' INSTRUCCIONES_FUNCION
                    | RVOID id ('')' INSTRUCCIONES_FUNCION
;


DECLARACION_FUNCION:   TIPO id '('PARAMETROS')' INSTRUCCIONES_FUNCION
                     | TIPO id ('')' INSTRUCCIONES_FUNCION
;


PARAMETROS:            PARAMETROS ',' DECLARACION_VACIA_METODO
                     | DECLARACION_VACIA_METODO
;


DECLARACION_VACIA_METODO:     TIPO  id
;


LLAMADA_METODOS:       id '('PARAMETROS_LLAMADA')'
                     | id ('')'
;
PARAMETROS_LLAMADA:    PARAMETROS_LLAMADA ',' EXPRESION
                     | EXPRESION
;
```

```
ASIGNACION_VARIABLE  :        id '=' EXPRESION
                             | id '++'
                             | id '--'
;
ASIGNACION_VECTORES :        id '[' EXPRESION ']' '=' EXPRESION ';'
;
ASIGNACION_LISTAS :          id '[''[' EXPRESION ']'']' '=' EXPRESION ';'
;


INSTRUCCIONES_FUNCION:        '{'instrucciones'}'
;



FUNCION_IF:                 RIF '(' EXPRESION ')'INSTRUCCIONES_FUNCION
                            | RIF '(' EXPRESION ')' INSTRUCCIONES_FUNCION
RELSE INSTRUCCIONES_FUNCION
                           | RIF '(' EXPRESION ')' INSTRUCCIONES_FUNCION
RELSE FUNCION_IF
;


FUNCION_PRINT:               RPRINT '('EXPRESION')'
;


FUNCION_WHILE:               RWHILE '(' EXPRESION ')'
INSTRUCCIONES_FUNCION
;


FUNCION_FOR:                 RFOR '(' DEC_O_ASIG ';' EXPRESION ';'
ASIGNACION_VARIABLE ')' INSTRUCCIONES_FUNCION
;


FUNCION_DO_WHILE:       RDO INSTRUCCIONES_FUNCION RWHILE'(' EXPRESION ')'
;
```

```
DEC_O_ASIG:                  ASIGNACION_VARIABLE
                           | DECLARACION_VARIABLE
;


FUNCION_SWITCH:    RSWITCH '(' EXPRESION ')' '{' CASES_SWITCH         '}'
                 | RSWITCH '(' EXPRESION ')' '{' CASES_SWITCH DEFAULT '}'
                 | RSWITCH '(' EXPRESION ')' '{'              DEFAULT '}'
;


CASES_SWITCH:     CASES_SWITCH CASE_SWITCH
                | CASE_SWITCH
;


CASE_SWITCH:      RCASE EXPRESION ‘:’ INSTRUCCIONES_SWITCH
;


DEFAULT:          RDEFAULT ':' INSTRUCCIONES_SWITCH
;


INSTRUCCIONES_SWITCH:     instrucciones
.                        |
;


SENTENCIA_BREAK:           RBREAK ';'
;
SENTENCIA_CONTINUE:        RCONTINUE ';'
;
SENTENCIA_RETURN:          RRETURN EXPRESION ';'
                         | RRETURN ';'
;
TIPO    :       RINT
        |       RBOOLEAN
        |       RSTRING
        |       RDOUBLE
        |       RCHAR
;
```

```
EXPRESION :   EXPRESION '+' EXPRESION
    |   EXPRESION '-' EXPRESION
    |   EXPRESION '*' EXPRESION
    |   EXPRESION '/' EXPRESION
    |   EXPRESION '^' EXPRESION
    |   EXPRESION '%' EXPRESION
    |   '-' EXPRESION %prec negativo
    |   '(' EXPRESION ')'
    |   EXPRESION '=='  EXPRESION
    |   EXPRESION '!='  EXPRESION
    |   EXPRESION '<'   EXPRESION
    |   EXPRESION '>'   EXPRESION
    |   EXPRESION '<='  EXPRESION
    |   EXPRESION '>='  EXPRESION
    |   EXPRESION '||'  EXPRESION
    |   EXPRESION '&&'  EXPRESION
    |   '!'  EXPRESION
    |   EXPRESION '++'
    |   EXPRESION '--'
    |   RTOLOWER '(' EXPRESION ')'
    |   RTOUPPER '(' EXPRESION ')'
    |   RLENGTH '(' EXPRESION ')'
    |   RTRUNCATE '(' EXPRESION ')'
    |   RROUND '(' EXPRESION ')'
    |   RTYPEOF '(' EXPRESION ')'
    |   RTOSTRING '(' EXPRESION ')'
    |   '(' RINT ')' EXPRESION
    |   '(' RDOUBLE ')' EXPRESION
    |   '(' RCHAR ')' EXPRESION
    |   '(' RSTRING ')' EXPRESION
    |   EXPRESION '?' EXPRESION ':' EXPRESION
    |   id '('PARAMETROS_LLAMADA')'
    |   id '('')'
    |   id '[''[' EXPRESION']'']'
    |   id
    |   id '['EXPRESION']'
    |   ENTERO
    |   DECIMAL
    |   CARACTER
    |   CADENA
    |   RTRUE
    |   RFALSE
    |   error
;
```

Descripcion de producciones

Producciones de DECLARACION_VARIABLES

• TIPO id '=' EXPRESION:
Genera un objeto tipo DECLARACION_VARIABLE al que se le envía el tipo, id y expresión, también genera su respectivo nodo de grafica

• TIPO id:
Genera un objeto de tipo DECLARACION_VARIABLE al que se le envía el tipo, id, y, también genera su respectivo nodo de grafica

Producciones de DECLARACION_VECTORES

• TIPO '[' ']' id '=' 'RNEW' TIPO '['EXPRESION']':
Genera un objeto de tipo DECLARACION_VECTOR_TIPO1, y su respectivo nodo de grafica

• TIPO '[' ']' id '=' '{' LISTA_EXPRESIONES '}':
Genera un objeto de tipo DECLARACION_VECTOR_TIPO1, y su respectivo nodo de grafica

Producciones de DECLARACION_LISTAS

• RLIST '<' TIPO '>' id '=' RNEW RLIST '<' TIPO '>':
Genera un objeto de tipo DECLARACION_LISTA_TIPO1, y su respectivo nodo de grafica

• RLIST '<' TIPO '>' id '=' EXPRESION:
Genera un objeto de tipo DECLARACION_LISTA_TIPO1, y su respectivo nodo de grafica

Producciones de DECLARACION_METODO

• RVOID id '('PARAMETROS')' INSTRUCCIONES_FUNCION
Genera un objeto de tipo DECLARACION_METODO al que se le envía el tipo, id, parámetros, e instrucciones. También genera su respectivo nodo de grafica

• RVOID id '('')' INSTRUCCIONES_FUNCION:
Genera un objeto de tipo DECLARACION_METODO al que se le envía el tipo, id, parámetros, e instrucciones. También genera su respectivo nodo de grafica

Producciones de LLAMADA_METODOS

id '('PARAMETROS_LLAMADA')':
Genera un objeto de LLAMADA_OBJETO el cual recibe como parámetros el id, y parámetros llamada, y genera su respectivo grafo.

id '(')':
Genera un objeto de LLAMADA_OBJETO el cual recibe como parámetros el id, y una lista vacía para indicar que no hay parámetros, y genera su respectivo grafo.

Producciones de ASIGNACION_VARIABLE

id '=' EXPRESION:
Genera un objeto de ASIGNACION_VARIABLE, el cual recibe como parámetros el id, y la expresión, también genera su respectivo grafo.

id '++':

Genera un objeto de OPERACIÓN_UNARIA, el cual recibe el operando y el id, También genera un objeto de ASIGNACION_VARIABLE el cual recibe como parámetros el id y el valor de la operación unaria. También genera su respectivo nodo grafica.

id '--':

Genera un objeto de OPERACIÓN_UNARIA, el cual recibe el operando y el id, También genera un objeto de ASIGNACION_VARIABLE el cual recibe como parámetros el id y el valor de la operación unaria. También genera su respectivo nodo grafica.

Producciones de ASIGNACION_VECTORES

id '[' EXPRESION ']' '=' EXPRESION ';':
Genera un objeto de ASIGNACION_VECTOR el cual recibe como parámetros el id, la posición para guardar, y la expresión que deseamos guardar, también genera su nodo de grafo

Producciones de ASIGNACION_LISTAS

id '['[' EXPRESION ']']' '=' EXPRESION ';':
Genera un objeto de tipo ASIGNACION_LISTA el cual recibe como parámetros el id de la lista, la posición y el valor que deseamos guardar, También genera su nodo de grafo.

Producciones de AGREGAR_DATO_LISTAS

id '.' RADD '(' EXPRESION ')':
Genera un objeto de AGREGAR_A_LISTA el cual recibe como parámetros el id de la lista,

y la expresión que deseamos añadir, también genera el nodo del grafo.

Producciones de FUNCION_IF

RIF '(' EXPRESION ')' INSTRUCCIONES_FUNCION:
Genera un objeto de tipo IF, el cual recibe como parámetros la expresión booleana, y la lista de instrucciones si es verdadera y una lista vacía para instrucciones si es falsa, también genera el nodo de grafica.

RIF '(' EXPRESION ')' INSTRUCCIONES_FUNCION RELSE INSTRUCCIONES_FUNCION
Genera un objeto de tipo IF, el cual recibe como parámetros la expresión booleana, y la lista de instrucciones si es verdadera y la lista de instrucciones si es falsa, también genera un nodo grafica

RIF '(' EXPRESION ')' INSTRUCCIONES_FUNCION RELSE FUNCION_IF

Genera un objeto de tipo IF, el cual recibe como parámetros la expresión booleana, y la lista de instrucciones si es verdadera y con un objeto IF en la lista si es falsa, también genera un nodo grafica

Producciones de FUNCION_PRINT

RPRINT '('EXPRESION')':
Genera un objeto de tipo PRINT, el cual recibe como parámetros la expresión. Y también genera el nodo de la gráfica.

Producciones de FUNCION_WHILE

RWHILE '(' EXPRESION ')' INSTRUCCIONES_FUNCION:
Genera un objeto de tipo WHILE, el cual recibe como parámetros la expresión booleana, y la lista de instrucciones a ejecutar. También genera el nodo de la gráfica.

Producciones de FUNCION_FOR

RFOR '(' DEC_O_ASIG ';' EXPRESION ';' ASIGNACION_VARIABLE ')' INSTRUCCIONES_FUNCION:
Genera un objeto y tipo FOR el cual recibe como parámetros la declaración de variable, la expresión booleana, la asignación de variable, y la lista de instrucciones para ejecutar

Producciones de FUNCION_DO_WHILE

RDO INSTRUCCIONES_FUNCION RWHILE'(' EXPRESION ')':
Genera un objeto de tipo DO_WHILE el cual recibe como parámetros las instrucciones del deshile, y la expresión para el do. También genera el nodo de la grafica

Producciones de SWITCH

RSWITCH '(' EXPRESION ')' '{' CASES_SWITCH '}'
Genera un objeto de tipo SWITCH y recibe como parámetros la expresión a analizar, y la

Lista de cases. También genera el nodo del grafo

RSWITCH '(' EXPRESION ')' '{' CASES_SWITCH DEFAULT '}'
Genera un objeto de tipo SWITCH y recibe como parámetros la expresión a analizar, la

Lista de cases y el objeto default. También genera el nodo del grafo

RSWITCH '(' EXPRESION ')' '{' DEFAULT '}'

Genera un objeto de tipo SWITCH y recibe como parámetros la expresión a analizar y el objeto default. También genera el nodo del grafo

Producciones de FUNCION_MAIN

RMAIN id '('PARAMETROS_LLAMADA')'
Genera un objeto de tipo MAIN, el cual recibe como parámetros el id, y la lista de parámetros para la función. También genera el nodo de la gráfica.

RMAIN id '('')'
RMAIN id '('PARAMETROS_LLAMADA')'
Genera un objeto de tipo MAIN, el cual recibe como parámetros el id, una lista vacía para la función. También genera el nodo de la gráfica.

Producciones de SENTENCIA_BREAK

RBREAK ';'
Crea un objeto break, el cual no recibe ningún parámetro, también genera su nodo de grafica.

Producciones de SENTENCIA_CONTINUE

RCONTINUE ';'
Crea un objeto SENT_CONTINUE, el cual no recibe ningún parámetro, también genera su nodo de grafica.

Producciones de SENTENCIA_RETURN

RRETURN EXPRESION ';':
crea un objeto RETURN, el cual recibe como parámetro la expresión a retornar

RRETURN ';'
Crea un objeto SENT_RETURN, el cual recibe como parámetro una cadena vacía para indicar que no se retornara nada.

Producciones de TIPO

RINT
crea un objeto de tipo Tipo, con valor int. También genera su nodo de grafica

RBOOLEAN
crea un objeto de tipo Tipo, con valor boolean. También genera su nodo de grafica

RSTRING
crea un objeto de tipo Tipo, con valor string. También genera su nodo de grafica

RDOUBLE
crea un objeto de tipo Tipo, con valor double. También genera su nodo de grafica

RCHAR
crea un objeto de tipo Tipo, con valor char. También genera su nodo de grafica

Producciones de EXPRESION

EXPRESION '+' EXPRESION
Genera un objeto de tipo OPERACIONES el cual recibe como parámetros la expresión 1, el signo de la operación, y la expresión 2. También genera el nodo de la grafica

EXPRESION '-' EXPRESION
Genera un objeto de tipo OPERACIONES el cual recibe como parámetros la expresión 1, el signo de la operación, y la expresión 2. También genera el nodo de la grafica

EXPRESION '*' EXPRESION
Genera un objeto de tipo OPERACIONES el cual recibe como parámetros la expresión 1, el signo de la operación, y la expresión 2. También genera el nodo de la grafica

EXPRESION '^' EXPRESION
Genera un objeto de tipo OPERACIONES el cual recibe como parámetros la expresión 1, el signo de la operación, y la expresión 2. También genera el nodo de la grafica

EXPRESION '%' EXPRESION
Genera un objeto de tipo OPERACIONES el cual recibe como parámetros la expresión 1, el signo de la operación, y la expresión 2. También genera el nodo de la grafica

'-' EXPRESION %prec negativo
Genera un objeto de tipo OPERACIÓN_UNARIA el cual recibe como parámetros la expresión , y el signo de la operación. También genera el nodo de la grafica

'(' EXPRESION ')'

No genera objeto, pero retorna el valor de la expresion.tambien genera un nodo de su grafica.

EXPRESION '==' EXPRESION
Genera un objeto de tipo OPERACIONES el cual recibe como parámetros la expresión 1, el signo de la operación, y la expresión 2. También genera el nodo de la grafica

EXPRESION '!=' EXPRESION
Genera un objeto de tipo OPERACIONES el cual recibe como parámetros la expresión 1, el signo de la operación, y la expresión 2. También genera el nodo de la grafica

EXPRESION '<' EXPRESION
Genera un objeto de tipo OPERACIONES el cual recibe como parámetros la expresión 1, el signo de la operación, y la expresión 2. También genera el nodo de la grafica

EXPRESION '>' EXPRESION
Genera un objeto de tipo OPERACIONES el cual recibe como parámetros la expresión 1, el signo de la operación, y la expresión 2. También genera el nodo de la grafica

EXPRESION '<=' EXPRESION
Genera un objeto de tipo OPERACIONES el cual recibe como parámetros la expresión 1, el signo de la operación, y la expresión 2. También genera el nodo de la grafica

EXPRESION '>=' EXPRESION
Genera un objeto de tipo OPERACIONES el cual recibe como parámetros la expresión 1, el signo de la operación, y la expresión 2. También genera el nodo de la grafica

EXPRESION '||' EXPRESION
Genera un objeto de tipo OPERACIONES el cual recibe como parámetros la expresión 1, el signo de la operación, y la expresión 2. También genera el nodo de la grafica

EXPRESION '&&' EXPRESION
Genera un objeto de tipo OPERACIONES el cual recibe como parámetros la expresión 1, el signo de la operación, y la expresión 2. También genera el nodo de la grafica

'!' EXPRESION
Genera un objeto de tipo OPERACIÓN_UNARIA el cual recibe como parámetros la expresión , y el signo de la operación. También genera el nodo de la grafica

EXPRESION '++'

Genera un objeto de tipo OPERACIÓN_UNARIA el cual recibe como parámetros la expresión , y el signo de la operación. También genera el nodo de la grafica

EXPRESION '--'

Genera un objeto de tipo OPERACIÓN_UNARIA el cual recibe como parámetros la expresión , y el signo de la operación. También genera el nodo de la grafica

RTOLOWER '(' EXPRESION ')'
Genera un objeto de tipo TO_LOWER el cual recibe como parámetros la expresión a utilizar. También genera el nodo de la grafica

RTOUPPER '(' EXPRESION ')'

Genera un objeto de tipo TO_UPPER el cual recibe como parámetros la expresión a utilizar. También genera el nodo de la grafica

RLENGTH '(' EXPRESION ')'

Genera un objeto de tipo LENGTH el cual recibe como parámetros la expresión a utilizar. También genera el nodo de la grafica

RTRUNCATE '(' EXPRESION ')'

Genera un objeto de tipo TRUNCATE el cual recibe como parámetros la expresión a utilizar. También genera el nodo de la grafica

RROUND '(' EXPRESION ')'

Genera un objeto de tipo ROUND el cual recibe como parámetros la expresión a utilizar. También genera el nodo de la grafica

RTYPEOF '(' EXPRESION ')'

Genera un objeto de tipo TYPEOF el cual recibe como parámetros la expresión a utilizar. También genera el nodo de la grafica

RTOSTRING '(' EXPRESION ')'

Genera un objeto de tipo TO_STRING el cual recibe como parámetros la expresión a utilizar. También genera el nodo de la grafica

'(' RINT ')' EXPRESION

Genera un objeto de tipo CASTEOS el cual recibe como parámetros la expresión a utilizar. También genera el nodo de la grafica

'(' RDOUBLE ')' EXPRESION

Genera un objeto de tipo CASTEOS el cual recibe como parámetros la expresión a utilizar. También genera el nodo de la grafica

'(' RCHAR ')' EXPRESION

Genera un objeto de tipo CASTEOS el cual recibe como parámetros la expresión a utilizar. También genera el nodo de la grafica

'(' RSTRING ')' EXPRESION

Genera un objeto de tipo CASTEOS el cual recibe como parámetros la expresión a utilizar. También genera el nodo de la grafica

EXPRESION '?' EXPRESION ':' EXPRESION

Genera un objeto de tipo OPERACIÓN_TERNARIA el cual recibe como parámetros la expresión a utilizar como condición, la expresion en caso de verdadero, y la expresion en caso de falso. También genera el nodo de la grafica

id '('PARAMETROS_LLAMADA')'

Genera un objeto de tipo LLAMADA_METODO_EXP el cual recibe como parámetros el id de la función y la lista de expresiónes a utilizar. También genera el nodo de la grafica

id '(")'

Genera un objeto de tipo LLAMADA_METODO_EXP el cual recibe como parámetros el id de la función y una lista vacia. También genera el nodo de la grafica

id '["[' EXPRESION']"]'

Genera un objeto de tipo VALIDAR_EXISTE_LISTA el cual recibe como parámetros el id de la función y expresión a utilizar. También genera el nodo de la grafica

RTOCHARARRAY '('EXPRESION')'

Genera un objeto de tipo TOCHARARRAY el cual recibe como parámetros la expresión a utilizar. También genera el nodo de la grafica

Id

Genera un objeto de tipo VALIDAR_EXISTE_VARIABLE el cual recibe como parámetros la expresión a utilizar. También genera el nodo de la grafica

id '['EXPRESION']'

Genera un objeto de tipo VALIDAR_EXISTE_VECTOR el cual recibe como parámetros el id del vector y la posición que queremos obtener. También genera el nodo de la grafica

ENTERO

Genera un objeto de tipo VALOR el cual recibe como parámetros la expresión a utilizar. También genera el nodo de la grafica

DECIMAL

Genera un objeto de tipo VALOR el cual recibe como parámetros la expresión a utilizar. También genera el nodo de la grafica

CARÁCTER

Genera un objeto de tipo VALOR el cual recibe como parámetros la expresión a utilizar. También genera el nodo de la grafica

CADENA

Genera un objeto de tipo VALOR el cual recibe como parámetros la expresión a utilizar. También genera el nodo de la grafica

RTRUE

Genera un objeto de tipo VALOR el cual recibe como parámetros la expresión a utilizar. También genera el nodo de la grafica

RFALSE

Genera un objeto de tipo VALOR el cual recibe como parámetros la expresión a utilizar. También genera el nodo de la grafica