



# Tecnológico de Monterrey

## Diseño de Compiladores Proyecto Final

Profesora: Elda G. Quiroga

Gerardo Galan Garzafox A00821196  
02 de Oct. 2021

## Tabla de Contenido

<b>DESCRIPCIÓN DEL PROYECTO</b>	<b>3</b>
Propósito y Alcance del Proyecto	3
Análisis de Requerimientos y Test Cases	3
Descripción del PROCESO	4
<b>DESCRIPCIÓN DEL LENGUAJE</b>	<b>4</b>
Nombre del lenguaje	4
Características del lenguaje	4
Listado de Errores	4
<b>DESCRIPCIÓN DEL COMPILADOR</b>	<b>5</b>
Descripción técnica.	5
Descripción del Análisis de Léxico	5
Descripción del Análisis de Sintaxis.	7
Análisis Semántico y Generación de Código Intermedio.	7
Administración de Memoria	8
<b>DESCRIPCIÓN DE LA MÁQUINA VIRTUAL</b>	<b>9</b>
Descripción técnica	9
Arquitectura	9
<b>PRUEBAS DEL FUNCIONAMIENTO DEL LENGUAJE</b>	<b>9</b>
<b>DOCUMENTACIÓN DEL CÓDIGO DEL PROYECTO</b>	<b>9</b>
<b>MANUAL DE USUARIO</b>	<b>9</b>
<b>LINKS DE REFERENCIA</b>	<b>9</b>

# 1. Calendario de proyecto

Semana	L	Mi	V	AVANCE	Contenido esperado de la entrega
Sep-Oct (27- 1)	27	29	1	#1	Análisis de Léxico y Sintaxis (Scanner y Parser)
Octubre (4 - 8)	4	6	8	#2	Semántica Básica de Variables: Directorio de Procedimientos y Tablas de Variables Semántica Básica de Expresiones: Tabla de Consideraciones semánticas (Cubo Semántico)
Octubre (11 - 15)	11	13	15	#3	Generacion de Código de Expresiones Aritméticas y estatutos secuenciales: Asignación, Lectura, etc. Generacion de Código de Estatutos Condicionales: Decisiones
Octubre (18 – 22)	18	20	22	#4	Generacion de Código de Estatutos Condicionales: Ciclos Generacion de Código de Funciones
Octubre (25 - 29)	25	27	29	---	<b>SEMANA - I</b>
Noviembre (1 – 5)	1	3	5	#5	Mapa de Memoria de Ejecución para la Máquina Virtual Máquina Virtual: Ejecución de Expresiones Aritméticas y Estatutos Secuenciales
Noviembre (8 – 12)	8	10	12	#6	Generacion de Código de Arreglos /Tipos estructurados Máquina Virtual: Ejecución de Estatutos Condicionales
Noviembre (15– 19)	15	17	19	#7	1era versión de la Documentación Generacion de Código y Máquina Virtual para una parte de la aplicación particular
Noviembre (22 - 26)	22	24	26	FINAL	<b>ENTREGA FINAL DEL PROYECTO Noviembre 24, 12:00pm</b>

Diagrama de Gantt del Proyecto PROGRAMACIÓN DE AVANZADO

## 2. COMPILADOR

### 2.1. Tokens

#	Token	Syntaxis
1	PROGRAM	program
2	MAIN	main
3	VARS	vars
4	INT	int

5	FLOAT	float
6	BOOL	bool
7	CHAR	char
8	STRING	string
9	FUNCTION	function
10	RETURN	return
11	READ	read
12	WRITE	write
13	IF	if
14	THEN	then
15	ELSE	else
16	WHILE	while
17	DO	do
18	FOR	for
19	TO	to
20	VOID	void
21	AND	and
22	OR	or
23	NOT	not
24	AND	&&
25	OR	
26	NOT	!
27	LESS	<
28	GREATER	>
29	LESS_EQ	<=
30	GREATER_EQ	>=
31	EQUIVALENT	==
32	DIFFERENT	!=
33	EQUAL	=

34	MULT	*
35	DIV	/
36	PLUS	+
37	MINUS	-
38	MULT_EQ	*=
39	DIV_EQ	/=
40	PLUS_EQ	+=
41	MINUS_EQ	-=
42	L_BRACE	{
43	R_BRACE	}
44	L_BRACKET	[
45	R_BRACKET	]
46	L_PAR	(
47	R_PAR	)
48	COLON	:
49	SEMICOLON	;
50	COMMA	,
51	ID	[a-zA-Z][a-zA-Z_0-9]*
52	MEDIAN	median\b
53	MODE	mode\b
54	MEAN	mean\b
55	VARIANCE	variance\b
56	REGRESSION	regression\b
57	PLOT_XY	plotXY\b
58	MAX	max\b
59	MIN	min\b
60	LENGTH	length\b
61	CTE_INT	-?\d+
62	CTE_FLOAT	-?\d+\.\d+

63	CTE_BOOL	(True   False   true   false)
64	CTE_CHAR	( \'\'   \'\' )
65	CTE_STRING	( \".+\"   \'.+\')
66	REMAINDER	%
67	EXP	**
68	EQUIVALENT	is\b
69	DIFFERENT	isnt\b

## 2.2. Gramatica Formal

<b>&lt;PROGRAM&gt;</b> A → program id ; B B → VARS C   C C → FUNCTION C   MAIN eps	<b>&lt;VARS&gt;</b> A → vars { B } B → C   CB C → TYPE : D ; D → VAR   VAR , D
<b>&lt;STATEMENTS&gt;</b> A → ASSIGN → CONDICIONAL → READ → WRITE → LOOP_COND → LOOP_RANGE → RETURN → VOID	<b>&lt;TYPE&gt;</b> A → int → float → char → bool → string
<b>&lt;RETURN&gt;</b> A → return ( EXP ) ;	<b>&lt;OPER-ASSIGN&gt;</b> A → +=   -=   *=   /=
<b>&lt;LOOP_RANGE&gt;</b> A → for VAR = EXP to EXP do BLOCK	<b>&lt;NOT&gt;</b> A → eps   !   not
A → id ( B ) ; B → EXPRESSION   EXPRESSION , B	<b>&lt;FUNC_TYPE&gt;</b> A → void   TYPE
<b>&lt;FUNCTION&gt;</b> A → function FUNC_TYPE id ( B ) BLOCK	<b>&lt;VAR&gt;</b> A → id

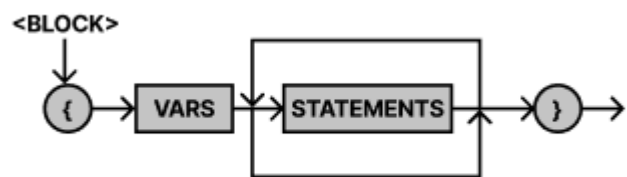
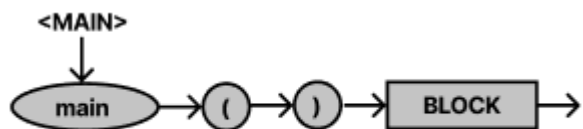
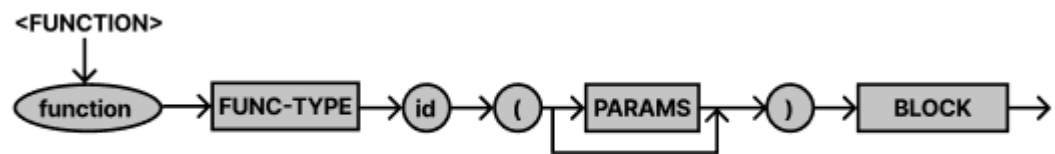
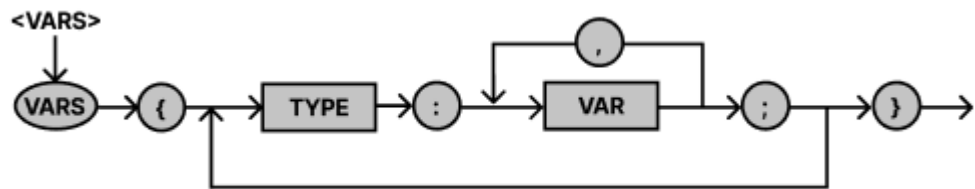
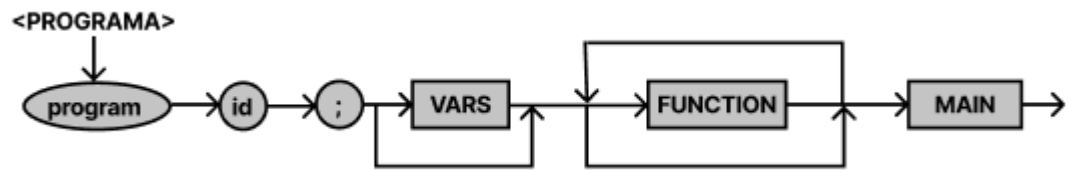
$B \rightarrow \text{PARAMS} \mid \text{eps}$	$\rightarrow \text{VECTOR}$
$\langle \text{BLOCK} \rangle$ $A \rightarrow B \{ C \}$ $B \rightarrow \text{VARS} \mid \text{eps}$ $C \rightarrow \text{STATEMENTS} \mid \text{STATEMENTS } B$	$\langle \text{PARAMS} \rangle$ $A \rightarrow B \mid B, A$ $B \rightarrow \text{TYPE} : \text{VAR}$
$\langle \text{CONDICIONAL} \rangle$ $A \rightarrow \text{if} ( \text{EXPRESSION} ) \text{ then } \text{BLOCK } B ;$ $B \rightarrow \text{else } \text{BLOCK} \mid \text{eps}$	$\langle \text{MAIN} \rangle$ $A \rightarrow \text{main} ( ) \text{ BLOCK}$
$\langle \text{LOOP\_COND} \rangle$ $A \rightarrow \text{while} ( \text{EXPRESSION} ) \text{ do } \text{BLOCK}$	$\langle \text{ASSIGN} \rangle$ $A \rightarrow \text{VAR } B \text{ } C ;$ $B \rightarrow = \mid \text{OPER\_ASSIGN}$ $C \rightarrow \text{EXPRESSION} \mid \text{FUNC} \mid \text{PREDEF\_FUNC}$
$\langle \text{WRITE} \rangle$ $A \rightarrow \text{write} ( B ) ;$ $B \rightarrow \text{EXPRESSION } C \mid \text{cte.string } C$ $C \rightarrow , B \mid \text{eps}$	$\langle \text{READ} \rangle$ $A \rightarrow \text{read} ( B ) ;$ $B \rightarrow \text{VAR} \mid \text{VAR}, B$
$\langle \text{LOGIC} \rangle$ $A \rightarrow \text{EXP } B$ $B \rightarrow \text{eps} \mid C \text{ EXP}$ $C \rightarrow < \mid \leq \mid > \mid \geq \mid == \mid != \mid \text{is} \mid \text{isnt}$	$\langle \text{EXPRESSION} \rangle$ $A \rightarrow \text{NOT } \text{LOGIC } B$ $B \rightarrow \text{eps} \mid C \text{ A}$ $C \rightarrow \text{or} \mid \text{"  "} \mid \&\& \mid \text{and}$
$\langle \text{TERM} \rangle$ $A \rightarrow \text{FACTOR } B$ $B \rightarrow \text{eps} \mid * A \mid / A \mid \% A$	$\langle \text{EXP} \rangle$ $A \rightarrow \text{TERM } B$ $B \rightarrow \text{eps} \mid - A \mid + A$
$\langle \text{VAR\_CTE} \rangle$ $A \rightarrow \text{id}$ $\quad \rightarrow \text{cte.int}$ $\quad \rightarrow \text{cte.float}$ $\quad \rightarrow \text{cte.char}$ $\quad \rightarrow \text{cte.string}$ $\quad \rightarrow \text{cte.bool}$	$\langle \text{EXPONENT} \rangle$ $A \rightarrow B \mid C$ $B \rightarrow ( \text{EXPRESSION} )$ $C \rightarrow D \text{ VAR\_CTE}$ $D \rightarrow \text{eps} \mid + \mid -$
$\langle \text{FUNC} \rangle$ $A \rightarrow \text{id} ( \text{EXPRESSION } B ) ;$ $B \rightarrow , \text{EXPRESSION } B \mid \text{eps}$	$\langle \text{VECTOR} \rangle$ $A \rightarrow \text{id} [ \text{cte\_int} ]$
$\langle \text{PREDEF\_FUNC} \rangle$ $A \rightarrow \text{length} ( \text{VECTOR} )$ $\quad \mid \text{max} ( \text{VECTOR} )$	$\langle \text{VOID} \rangle$ $A \rightarrow \text{FUNC} ;$

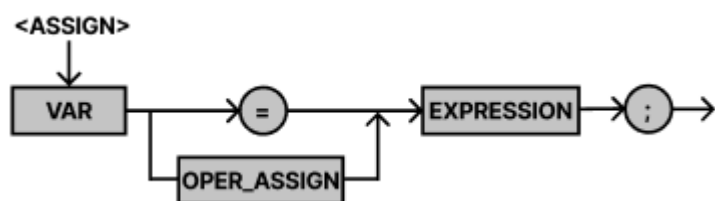
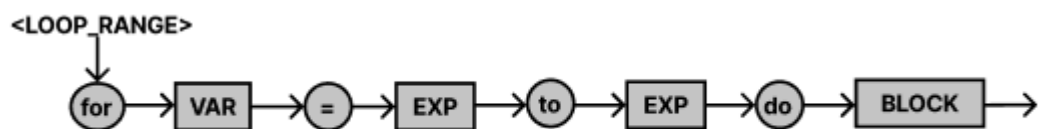
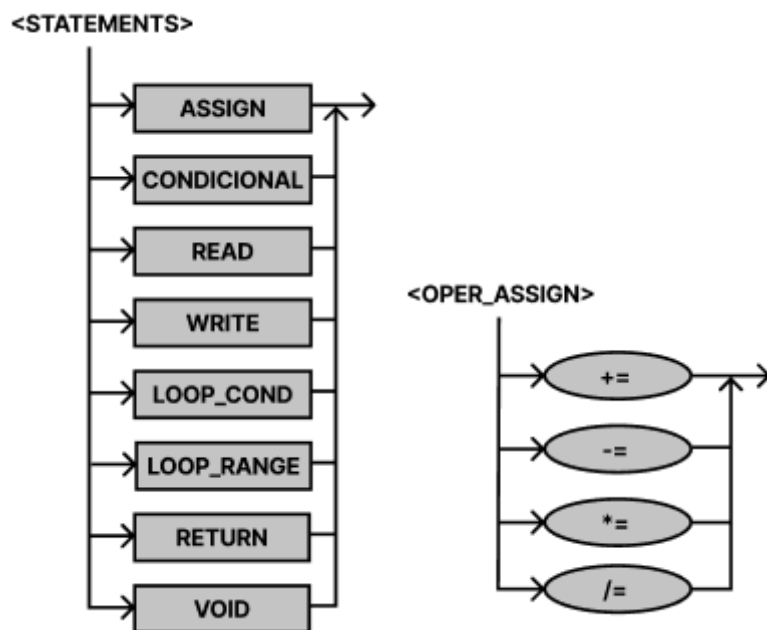
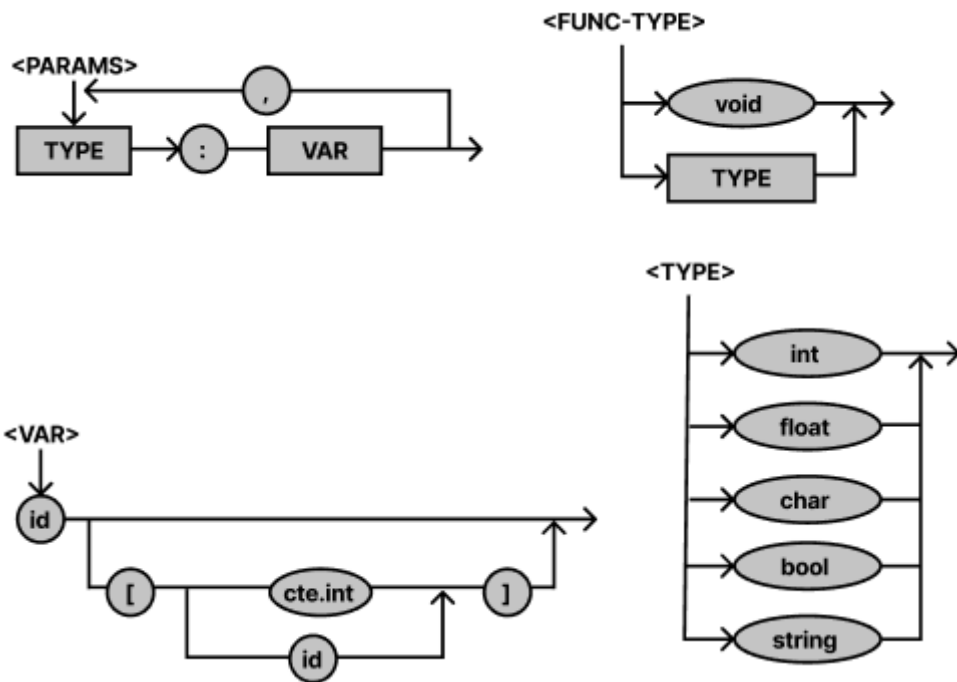
min ( VECTOR )   mean ( VECTOR )   median ( VECTOR )   mode ( VECTOR )   plotXY ( VECTOR, VECTOR )   regression ( VECTOR, VECTOR )   variance (VECTOR )   dot(VECTOR, VECTOR)   abs(cte_int)   sum(VECTOR)   roof(cte_float)   floor( cte_float)	
<FACTOR> A →EXPONENT   EXPONENT B B →** A	

## 2.3 Diagrama de Sintaxis y acciones semánticas

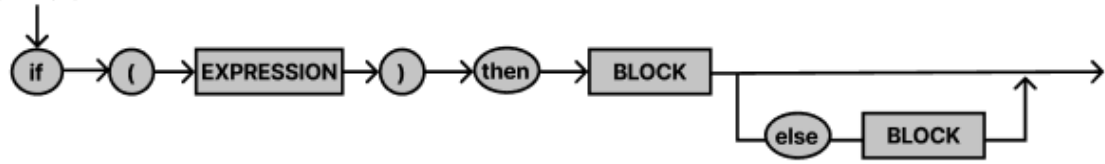
<https://www.figma.com/file/rC7N20A0q4SngvwEQAkQV0/Diagama-de-sintaxis?node-id=0%3A1>



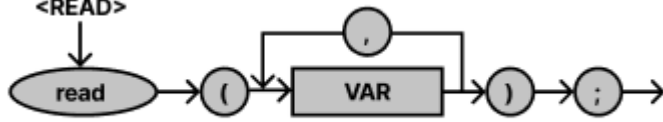




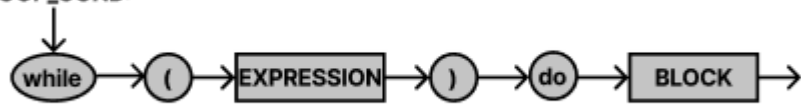
<CONDICIONAL>



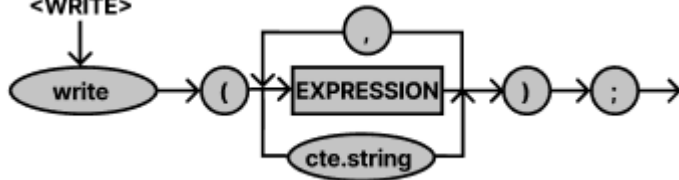
<READ>



<LOOP\_COND>



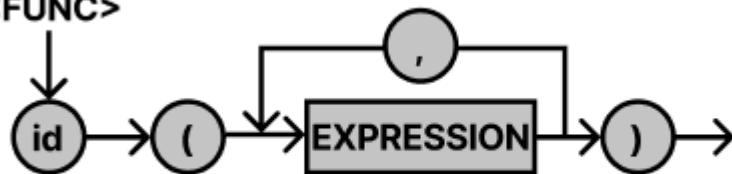
<WRITE>

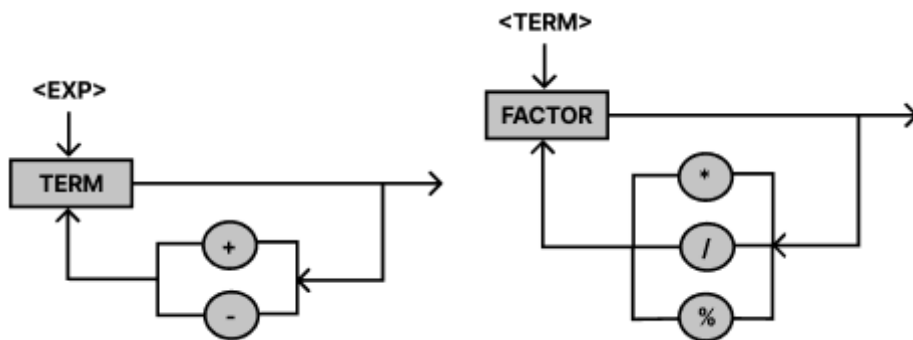
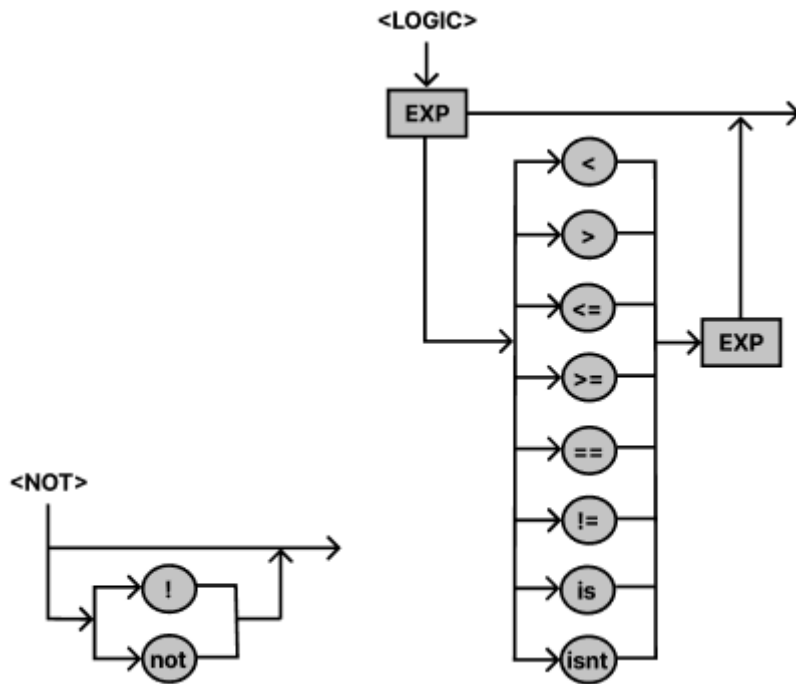
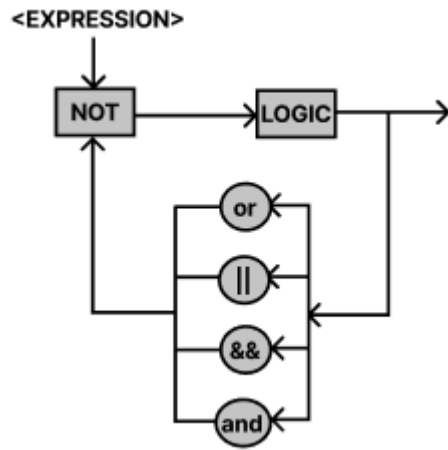
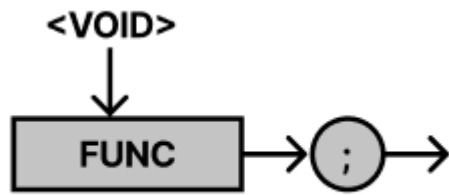


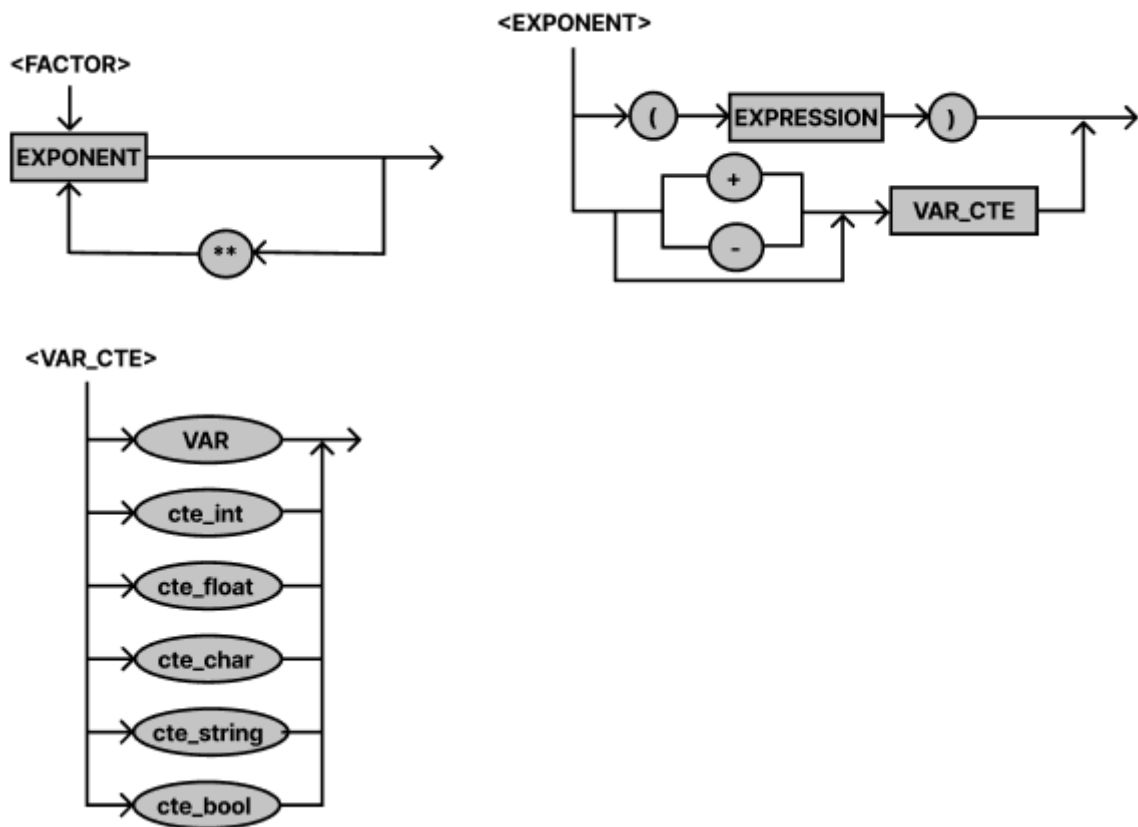
<RETURN>



<FUNC>







- Jerarquia de Operadores

Jerarquia
**
*
/
+
-
<
>
<=
>=
==
!=
&&

and
or
!
not
+=
-=
*=
/=

1. Tabla de consideraciones semánticas (cubo semántico)

(<,>,<=,>=)		Right Operator				
Left Operator		INT	FLOAT	CHAR	BOOL	STRING
	INT	BOOL	BOOL	error	error	error
	FLOAT	BOOL	BOOL	error	error	error
	CHAR	error	error	error	error	error
	BOOL	error	error	error	error	error
	STRING	error	error	error	error	error

(and, or, &&,   )		Right Operator				
Left Operator		INT	FLOAT	CHAR	BOOL	STRING
	INT	error	error	error	error	error
	FLOAT	error	error	error	error	error
	CHAR	error	error	error	error	error
	BOOL	error	error	error	BOOL	error
	STRING	error	error	error	error	error

(!=, ==)	Right Operator					
Left Operator		<b>INT</b>	<b>FLOAT</b>	<b>CHAR</b>	<b>BOOL</b>	<b>STRING</b>
	<b>INT</b>	BOOL	BOOL	BOOL	BOOL	BOOL
	<b>FLOAT</b>	BOOL	BOOL	BOOL	BOOL	BOOL
	<b>CHAR</b>	BOOL	BOOL	BOOL	BOOL	BOOL
	<b>BOOL</b>	BOOL	BOOL	BOOL	BOOL	BOOL
	<b>STRING</b>	BOOL	BOOL	BOOL	BOOL	BOOL

(* ,*=)	Right Operator					
Left Operator		<b>INT</b>	<b>FLOAT</b>	<b>CHAR</b>	<b>BOOL</b>	<b>STRING</b>
	<b>INT</b>	INT	FLOAT	STRING	error	STRING
	<b>FLOAT</b>	FLOAT	FLOAT	error	error	error
	<b>CHAR</b>	STRING	error	error	error	error
	<b>BOOL</b>	error	error	error	error	error
	<b>STRING</b>	STRING	error	error	error	error

(+, +=)	Right Operator					
Left Operator		<b>INT</b>	<b>FLOAT</b>	<b>CHAR</b>	<b>BOOL</b>	<b>STRING</b>
	<b>INT</b>	INT	FLOAT	error	error	error
	<b>FLOAT</b>	FLOAT	FLOAT	error	error	error
	<b>CHAR</b>	error	error	STRING	error	STRING
	<b>BOOL</b>	error	error	error	error	error
	<b>STRING</b>	error	error	STRING	error	STRING

(-, -=, **)	Right Operator					
Left		<b>INT</b>	<b>FLOAT</b>	<b>CHAR</b>	<b>BOOL</b>	<b>STRING</b>
	<b>INT</b>	INT	FLOAT	error	error	error
	<b>FLOAT</b>	FLOAT	FLOAT	error	error	error

Operator	<b>CHAR</b>	error	error	error	error	error
	<b>BOOL</b>	error	error	error	error	error
	<b>STRING</b>	error	error	error	error	error

(/,/=)	Right Operator					
Left Operator		<b>INT</b>	<b>FLOAT</b>	<b>CHAR</b>	<b>BOOL</b>	<b>STRING</b>
	<b>INT</b>	FLOAT	FLOAT	error	error	error
	<b>FLOAT</b>	FLOAT	FLOAT	error	error	error
	<b>CHAR</b>	error	error	error	error	error
	<b>BOOL</b>	error	error	error	error	error
	<b>STRING</b>	error	error	error	error	error