



Instituto Tecnológico y de Estudios Superiores de Monterrey

**TC3002B Desarrollo de Aplicaciones avanzadas de Ciencias Computacionales Gpo 502
Patito – Entrega #0**

José Eduardo de Valle Lara

A01734957

Profesora:

Elda Guadalupe Quiroga González

Tokens y Expresiones Regulares

Se agruparon los tokens con sus respectivas expresiones regulares en una misma sección para simplificar el documento. Las expresiones regulares de las palabras reservadas y algunos operadores son el mismo token ya que son patrones que no cambian, son fijos.

Token	Expresión Regular	Explicación
ID	[a-zA-Z][a-zA-Z0-9]*	Se usa para nombrar variables y funciones, debe comenzar con una letra y puede ser seguido por cualquier combinación de letras y números
Constantes: valores fijos que no cambian durante la ejecución del programa		
Enteros (cte_int)	[0-9]+	Debe haber al menos un número
Flotantes (cte_float)	[0-9]+\.[0-9]+	Empiezan con al menos uno o muchos números seguido de un punto "." y de al menos un número después del punto
Strings (cte.string)	"[^"]*"	Empiezan con una comilla doble " seguida de cero o más repeticiones de cualquier caracter que no sea una comilla doble y finaliza con una comilla doble
Operadores y símbolos: se usan para realizar operaciones, comparaciones y agrupaciones		
+	\+	Se usó un backslash \ para indicar la expresión regular de + ya que en expresiones regulares se refiere a uno o más elementos
-	-	
*	*	Se usó un backslash \ para indicar la expresión regular de * ya que en expresiones regulares se refiere a cero o más elementos
/	/	
>	>	
<	<	
!=	!=	

=	=	
;	;	
:	:	
,	,	
(\(Se usó un backslash \ para indicar la expresión regular de (porque se usa para agrupaciones en expresiones regulares
)	\)	Se usó un backslash \ para indicar la expresión regular de) porque se usa para agrupaciones en expresiones regulares
{	\{	Se usó un backslash \ para indicar la expresión regular de { porque se usa para cuantificadores en expresiones regulares
}	\}	Se usó un backslash \ para indicar la expresión regular de } porque se usa para cuantificadores en expresiones regulares

Palabras reservadas: identificadores con un significado especial que no pueden usarse como nombres de variables o funciones

program	program	
main	main	
end	end	
var	var	
int	int	
float	float	
void	void	
if	if	
else	else	
while	while	
do	do	

print	print	
-------	-------	--

Definición de la gramática libre de contexto

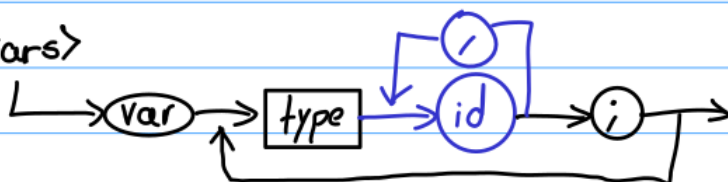
$\langle \text{Programa} \rangle$

$\langle \text{Programa} \rangle \rightarrow \text{program id ; } \langle \text{Opt_vars} \rangle \langle \text{Opt_funcs} \rangle \text{ main } \langle \text{Body} \rangle \text{ end}$

$\langle \text{Opt_vars} \rangle \rightarrow \langle \text{Vars} \rangle \mid \epsilon$

$\langle \text{Opt_funcs} \rangle \rightarrow \langle \text{FUNCS} \rangle \mid \epsilon$

$\langle \text{Vars} \rangle$



$\langle \text{Vars} \rangle \rightarrow \text{var } \langle \text{DEC_VAR} \rangle$

$\langle \text{DEC_VAR} \rangle \rightarrow \langle \text{TYPE} \rangle \langle \text{LIST_id} \rangle ; \langle \text{DV}' \rangle$

↑
puede tener 1
o más ids

↑
puede tener un
ciclo o no

$\langle \text{DV}' \rangle \rightarrow \langle \text{DEC_VAR} \rangle \mid \epsilon$

$\langle \text{FUNCS} \rangle$

$\langle \text{FUNCS} \rangle \rightarrow \langle \text{FUNC_DEF} \rangle$

$\langle \text{FUNC_DEF} \rangle \rightarrow \langle \text{RETURN_TYPE} \rangle \text{id} (\langle \text{PARAMS} \rangle) \langle \text{OPTIONAL_LOCAL_VARS} \rangle$
 $\langle \text{Body} \rangle ;$

$\langle \text{RETURN_TYPE} \rangle \rightarrow \text{void} \mid \langle \text{TYPE} \rangle$

$\langle \text{PARAMS} \rangle \rightarrow \langle \text{PARAM_LIST} \rangle \mid \epsilon$

$\langle \text{PARAM_LIST} \rangle \rightarrow \text{id} : \langle \text{TYPE} \rangle \langle \text{MORE_PARAMS} \rangle$

$\langle \text{MORE_PARAMS} \rangle \rightarrow , \langle \text{PARAM_LIST} \rangle \mid \epsilon$

$\langle \text{OPTIONAL_LOCAL_VARS} \rangle \rightarrow \langle \text{VARS} \rangle \mid \epsilon$

$\langle \text{Body} \rangle$

$\langle \text{Body} \rangle \rightarrow \{ \langle \text{STATEMENT_LIST} \rangle \}$

$\langle \text{STATEMENT_LIST} \rangle \rightarrow \langle \text{STATEMENT} \rangle \langle \text{STATEMENT_LIST} \rangle \mid \epsilon$

$\langle \text{TYPE} \rangle$

$\langle \text{TYPE} \rangle \rightarrow \text{int} \mid \text{float}$

$\langle \text{STATEMENT} \rangle$

Esta regla presenta una ambigüedad en $\langle \text{ASSIGN} \rangle$ y $\langle \text{F_call} \rangle$ ya que las 2 invocan un id, el primero es para asignar variables y el segundo es para funciones, se resolvió con el factor común izquierdo

$\langle \text{STATEMENT} \rangle \rightarrow \text{id} \langle \text{ID_STATEMENT_CONTINUATION} \rangle \mid \langle \text{CONDITION} \rangle$
 $\mid \langle \text{CYCLE} \rangle \mid \langle \text{Print} \rangle$

$\langle \text{ID_STATEMENT_CONTINUATION} \rangle \rightarrow = \langle \text{EXPRESION} \rangle ; \mid (\langle \text{EXPRESION} \rangle) ;$

<ASSIGN>

<ASSIGN> \rightarrow id = <EXPRESSION>;

<CONDITION>

<CONDITION> \rightarrow if (<EXPRESSION>) <Body> <OPTIONAL_ELSE>;

<OPTIONAL_ELSE> \rightarrow else <Body> | ϵ

<CYCLE>

<CYCLE> \rightarrow while (<EXPRESSION>) do <Body>

$\langle F_call \rangle$

$\langle F_call \rangle \rightarrow id (\langle EXPRESSION \rangle);$

$\langle Print \rangle$

$\langle Print \rangle \rightarrow print (\langle PRINT_LIST \rangle);$

$\langle PRINT_LIST \rangle \rightarrow \langle PRINT_ITEM \rangle \langle PRINT_MORE_ITEMS \rangle$

$\langle PRINT_ITEM \rangle \rightarrow \langle EXPRESSION \rangle \mid cte.string$

$\langle PRINT_MORE_ITEMS \rangle \rightarrow , \langle PRINT_ITEM \rangle \langle PRINT_MORE_ITEMS \rangle \mid \epsilon$

$\langle EXPRESSION \rangle$

$\langle EXPRESSION \rangle \rightarrow \langle EXP \rangle \langle OPTIONAL_COMPARISON \rangle$

$\langle OPTIONAL_COMPARISON \rangle \rightarrow \langle RELATIONAL_OPERATOR \rangle \langle EXP \rangle \mid \epsilon$

$\langle RELATIONAL_OPERATOR \rangle \rightarrow > \mid < \mid !=$

$\langle EXP \rangle$

$\langle EXP \rangle \rightarrow \langle TERMINO \rangle \langle EXP_PRIME \rangle$

$\langle EXP_PRIME \rangle \rightarrow + \langle TERMINO \rangle \langle EXP_PRIME \rangle \mid - \langle TERMINO \rangle \langle EXP_PRIME \rangle \mid \epsilon$

$\langle TERMINO \rangle$

$\langle TERMINO \rangle \rightarrow \langle FACTOR \rangle \langle PRIME_TERM \rangle$

$\langle PRIME_TERM \rangle \rightarrow * \langle FACTOR \rangle \langle PRIME_TERM \rangle$

$\mid / \langle FACTOR \rangle \langle PRIME_TERM \rangle \mid \epsilon$

<FACTOR>

<FACTOR> \rightarrow (<EXPRESSION>) | id | <CTE>

<CTE>

<CTE> \rightarrow cte_int | cte_float