Procesadores de Lenguajes

MEMORIA FINAL

Grupo 55

Daniel Tomás Sánchez Aarón Cabero Blanco Alejandro Cuadrón Lafuente

 $Curso\ 2020/2021$

Índice

1	Intr	oducción	2
2	Disc	ño Analizador Léxico	3
	2.1	Tokens	 3
	2.2	Gramática Regular	 4
	2.3	Autómata Finito Determinista	 4
	2.4	Acciones Semánticas	 5
	2.5	Errores	 6
3	Disc	ño Analizador Sintáctico	7
	3.1	Gramática	 7
	3.2	Tabla LR(1)	 8
4	Disc	ño Analizador Semántico	11
5	Disc	ño Tabla de Símbolos	16
6	Ane	xo	17
	6.1	Casos de prueba correctos	 17
		6.1.1 Prueba correcta 1	 17
		6.1.2 Prueba correcta 2	 19
		6.1.3 Prueba correcta 3	 21
		6.1.4 Prueba correcta 4	 23
		6.1.5 Prueba correcta $5 \dots \dots \dots \dots \dots \dots \dots \dots \dots$	 26
	6.2	Casos de prueba erróneos	 30
		6.2.1 Prueba errónea 1	 30
		6.2.2 Prueba errónea 2	 30
		6.2.3 Prueba errónea 3	 30
		6.2.4 Prueba errónea 4	30
		6.2.5 Prueba errónea 5	 31
7	Ref	rancias	32

1 Introducción

El trabajo completo, tanto el léxico, como el sintáctico y el semático, ha sido realizado con la herramienta o librería externa "SLY"[1].

Opciones de grupo:

- Sentencias: Sentencia repetitiva (for)
- o Operadores especiales: Post-auto-decremento (-- como sufijo)
- o Técnicas de Análisis Sintáctico: **Ascendente**
- ∘ Comentarios: Comentario de bloque (/* */)
- Cadenas: Con comillas dobles (" ")

2 | Diseño Analizador Léxico

2.1 Tokens

 Identificador 	<ID, punteroTS $>$
■ Constante entera	$<\! {\rm CTEENTERA,\ valor} \! >$
■ Cadena de caracteres	<CADENA, lexema $>$
■ false	<CTELOGICA, $0>$
• true	<CTELOGICA, 1 $>$
 Palabra reservada Number 	<NUMBER, ->
 Palabra reservada String 	<STRING, - $>$
 Palabra reservada Boolean 	<BOOLEAN, ->
 Palabra reservada Let 	<LET $,$ - $>$
 Palabra reservada Alert 	<ALERT, - $>$
 Palabra reservada Input 	<INPUT, - $>$
 Palabra reservada Function 	<FUNCTION, - $>$
 Palabra reservada Return 	<RETURN, - $>$
 Palabra reservada If 	<if, -=""></if,>
 Palabra reservada For 	<FOR, - $>$
-	$\langle \text{OPESP}, - \rangle$
-	<OPARIT $,$ - $>$
■ =	<OPASIG, - $>$
■ ==	<OPREL, - $>$
& &	<OPLOG, - $>$
• (<ABPAREN, ->
	<CEAPAREN, $>$
• {	<ABLLAVE, ->
• }	<CELLAVE, - $>$
• ,	<COMA $,$ - $>$
. ;	<PUNTOYCOMA, ->

2.2 Gramática Regular

```
 \begin{array}{l} {\rm Axioma} = {\rm A} \\ {\rm A} \to {\rm del} \; {\rm A} \; | \; {\rm d} \; {\rm D} \; | \; " \; \; {\rm S} \; | \; / \; {\rm C} \; | \; {\rm I} \; {\rm I} \; | \; - {\rm M} \; | \; = \; {\rm E} \; | \; \& \; {\rm N} \; | \; (\; |\; ) \; | \; \{\; |\; \} \; | \; ; \; | \; , \\ {\rm D} \to {\rm d} \; {\rm D} \; | \; {\rm \lambda} \\ {\rm S} \to " \; | \; {\rm c} \; {\rm S} \\ {\rm C} \to * \; {\rm C} " \\ {\rm C} \to * \; {\rm C} " \; | \; {\rm c} \; {\rm C} " \\ {\rm C} " \to / \; {\rm A} \; | \; {\rm c} \; {\rm C} " \\ {\rm C} " \to / \; {\rm A} \; | \; {\rm c} \; {\rm C} " \\ {\rm I} \to {\rm d} \; {\rm I} \; | \; {\rm I} \; | \; | \; {\rm L} \; | \; {\rm A} \\ {\rm M} \to - \; | \; {\rm \lambda} \\ {\rm E} \to = \; | \; {\rm \lambda} \\ {\rm N} \to \& \\ \end{array}
```

Siendo d un dígito, l una letra, c cualquier otro carácter y del un delimitador.

2.3 Autómata Finito Determinista

2.4 Acciones Semánticas

```
A: leer
B: number = int(d), leer
C: number = number * 10 + int(d), leer
D: if number > 32767
    pError("Número fuera de rango")
  else
    genToken(CTEENTERA, number);
E: string = "", contador = 0, leer
F: string = string + otroCS, contador++, leer
G: if contador > 64
    pError("Cadena demasiado larga")
  else
    genToken(CADENA, string)
  leer
H: string = l, leer
I: string = string + l/D/_, leer
J: if palabrasReservadas.contains(string)
    if string == "number"
      genToken(NUMBER, -)
    elif string == "string"
      genToken(STRING,-)
    elif string == "boolean"
      genToken(BOOLEAN, -)
    elif string == "let"
      genToken(LET, -)
    elif string == "alert"
      genToken(ALERT, -)
    elif string == "input"
      genToken(INPUT, -)
    elif string == "return"
      genToken(RETURN, -)
    elif string == "if"
      genToken(IF, -)
    else
      genToken(FOR, -)
```

```
// palabrasReservadas.contains(string) = False
    puntero = TS.get(string)
    if zona decl == True
      if puntero!= None
        pError("Identificador ya declarado")
      else
        TS.update(string)
        puntero = TS.get(string)
        genToken(ID, puntero)
    else
      if puntero == None
        TS.update(string)
        puntero = TS.get(string)
        genToken(ID, puntero)
      else
        genToken(ID, puntero)
L: genToken(OPARIT, -)
M: genToken(OPESP, -), leer
N: genToken(OPASIG, -)
O: genTokeN(OPREL, -), leer
P: genToken(OPLOG, -), leer
Q: genToken(ABPAREN, -), leer
R: genToken(CEPAREN, -), leer
S: genToken(ABLLAVE, -), leer
T: genToken(CELLAVE, -), leer
U: genToken(COMA, -), leer
V: genToken(PUNTOYCOMA, -), leer
W: genToken(EOF, -), leer
```

2.5 Errores

Error léxico (siempre se lanza cuando el analizador léxico encuentra un error).

- 1. Cadena con longitud mayor de 64 caracteres.
- 2. Número fuera de rango (mayor de 32767).
- 3. Identificador ya declarado.
- 4. Carácter ilegal.

Todo error va acompañado de la *linea* y columna en el que se ha encontrado dicho error.

3 Diseño Analizador Sintáctico

3.1 Gramática

Axioma = B

```
No Terminales = { A B C D E F G H I J K L M N O P Q R S T U V W F1 F2 F3 }
Terminales = \{\&\& == - -- () = , ; \text{ id ent cad log let alert input return for if number } \}
                       boolean string function }
Producciones = {
                                                                               O \rightarrow \lambda
             B \to D
             D \to F D
                                                                               C \to G C
             \mathrm{D} \to \mathrm{G}\;\mathrm{D}
                                                                               C \to \lambda
             \mathrm{D} \to \lambda
                                                                               F \to F1~F2~F3
             G \rightarrow if (E) S
                                                                               F1 \rightarrow function P Q id
             \mathrm{G} \to \mathrm{S}
                                                                               P \rightarrow \lambda
             S \rightarrow H;
                                                                               Q \to T
             H \rightarrow id (I)
                                                                               Q \to \lambda
             I \to E \; J
                                                                               F2 \rightarrow (A)
                                                                               \mathbf{A} \to \mathbf{T}id AA
             I \rightarrow \lambda
             J \rightarrow , E J
                                                                               A \rightarrow \lambda
             J \to \lambda
                                                                               AA \rightarrow T id AA
             S \to K;
                                                                               AA \rightarrow \lambda
                                                                               F3 \rightarrow C
             K \rightarrow id = E
             S \rightarrow alert (E);
                                                                               E \rightarrow E \&\& R
             S \rightarrow input (id);
                                                                               E \to R
             S \rightarrow return L;
                                                                               R \rightarrow R == U
                                                                               \mathrm{R} 	o \mathrm{U}
             L \to E
                                                                               U \rightarrow U - V
             L \to \lambda
                                                                               \mathrm{U} \to \mathrm{V}
             G \rightarrow let M T id ;
             M \to \lambda
                                                                               V \rightarrow -- id
             T \rightarrow number
                                                                               V \rightarrow id
             T \to boolean
                                                                               V \rightarrow (E)
             T \rightarrow string
                                                                               V \to H
             G \rightarrow for (N; E; O) C
                                                                               V \rightarrow ent
             N \to K
                                                                               V \to cad
             N \to \lambda
                                                                               V \to \log
             O \to K
             O \rightarrow -- id
}
```

3.2 Tabla LR(1)

																			LR tabl	Le .																							
te	-1-4-	[ie]	, ,		1 64		ı _	la la mi	l fanns	luo tuu	m[]at	A	CTION	lated no	- Ean	,	, [16	an at i an l	ee 1			nt cad lo	- 0	D D	I c I	0	и Гт		77		w	- X	GOTO		ছি ছিৰ	la la	امطام	TA TA	x lpo		D [77
34	nbaa	36	()	+	s16	,	-	s13	s14	s15	n let	number	Doolean	string	s9	1	, .	- 1	10	88		- e	ent cad 10	9 5	1	3 7	7 1	1	U	12	L	M	I N	-		2 5	P	0 12	AA	A F3	E	R	-
s4 s4		s6 s6		-	s16			s13 s13	s14 s14	s15	88			-	a9 a9		<u>-</u>		10	-	_	-		+	17	3 7				12	+	+	+	╁	Н	2 5	H	╫	H	+	H	7	4
																	一	T		T				r ₃			Ť					T	┰	╈	П		m	т		\neg			T
			320																							Ţ												19					
r.	-		321	-	r-	₩	-	r.	Tr.	r.	7.		-	-	r-			-		-		-		-	₩	+	- -	-		_	-	-	-	╄	H		H	-H	-			-	
r ₅		rs		-	rs	₩	-	r ₅	rs	r ₅	r ₅			-	r ₅			-	5		-	-		+	₩	╁┼	-	-	-		- 2	2	-	+	Н	H	H	╌	-				-
			324						İ	İ																																	
s26	6	\Box		-07					_		\perp						[_	4							ж	П	4				4	4		_			25			\blacksquare	=		
+	-	H		327 328		┢	-	₩	+	-	+-			-	-			-			-	-		-	₩	╁	-	-	-	-	-	+	-	+	Н	H	H	╌	-				-
			329																																								
			30		-00				_		\perp														ж	П						4		_						\blacksquare	32 3		
533	,		839 844	-	±38	╁	845	-	+-	-	+			-			53	7		-	-	2	41 542 54	3	₩	₩	- 1	0		-	31	+	╬	╁	H	H	H	╫	H	+	32 3	34 3	35
		m				m			ÌΠ	İ	Ť				Т		一	T						rı	m	T	T							Ť									
																								r ₂																			
850	_	H		-	_	<u> </u>	_	<u> </u>	₩	-	+	-51	852	853	_	s47		4		_	_	_		-	₩	₩	_	_	-	_	-	4	_	-	ш	Щ.	щ		48	46	\rightarrow	_	_
850	,		360	+	s59	╁	-	╁	+-	-	╁	351	352	353	1		8	8	-	-	-	9	62 863 86	4	₩	╁	6	1		-	-	7	9	╁	Н	\vdash	H	-H	70	-	54 5	55	56
													s52	s53											П							6	5										
		Щ								\perp		r ₂₀	r ₂₀	r ₂₀				_[\Box			F	Щ	Щ								F	Ш		Щ	Щ	Į.	П	\Box	Ţ	
s68		H		-	s69	-		-	-	-	-	s51	852	s53				-			_			+	₩	+	_	-		67	-	7	66	8	H		-	70	H	#	4	-	-
r ₃₄		H		_								r ₃₄	r ₃₄	r ₃₄			-	7							#	H	+				7	- 1			Н		H	Н		H		-	
r ₆		r ₆			r ₆			r ₆		r ₆	re				r ₆			1	6					T	П									İ	П					T			
r ₁₂		r ₁₂			r ₁₂			r ₁₂	r ₁₂	r ₁₂	r ₁₂				r ₁₂			2	12																								
		2	360	-	s59 =74					H	-						35	8			_	9	62 863 86	4	F	H	6	1			4	F		F		H	H	4	Æ	Ŧ	73 5	55	56
		1		875	574						1						-	-							#	+	-				-	-		-	Н	-	H	H	r	Н		-	-
				r ₁₇															į.	376																							
				r ₁₈																																							
	_	Щ		r44		_			<u> </u>								_	_		44				_	Щ.	Щ					_	4		_	Ш	Щ	Щ	Щ	4		4	_	_
-	_	뉴		r46	-	┈	_	┡	₩	-	+	_		-	-	_	<u>-</u> -	4			46			-	₩	+	-		-	_	-	4	-	╄	Н	\vdash	H		-		-	-	¦
+	-	₩	-	r ₄₈	879	₩	-	₩	-	-	+		-	-				-	-	48	48	48		-	₩	₩		-	-		-	-	-	╁	Н	H	H	-H	-	-11		-	-
Ť	\neg		880	r50		一		\vdash	Ť	i –				i –	т		一	T		r ₅₀	r ₅₀	r ₅₀		T	₩	m	寸	_			7	T	┰	┰	т	H	m	┰		77	and the	\neg	T
÷	=		360	_	859	는	1	H	i	ì	i	=					3	88	i	=i	=i		62 863 86	4	ΤĖ	Τï	6	1		=i	=i	=	===	⇌	声	Ħ		Ħ	FF	Ŧ	81	55	56
Ť		m		r ₅₂		Ì		Ť	Ϊ		Ť			Ť –		i		T		52	r ₅₂	r ₅₂	T	Ť	ŤΓ	'nΪ	T	T	T	T	Tì	Ť	T	Ť	П	ΪT	m			7	П		
				r ₅₃																r ₅₃	r ₅₃	r ₅₃																					
	_	Щ		r ₅₄																	r ₅₄			Ļ		Ц	_		Ш	_	Ц	Ţ	Ļ										
-0.4	_	Щ.	-00	rss	-00	-		-	-	-	-	_		 	_			10		r _{SS}	r _{SS}		-00 -00 -0	+	-	щ		1 00	\blacksquare	_	_	4	+	╄	Н	Щ.	Щ		#	#	0.0	0.5	0.0
s84			90 839	+	s89 s38	₩		₩	-	-	+			+			5	37		-	-		92 593 59 941 542 54		₩	H		1 82	Н	-	-	+	+	╬	H	₩	H	₩	H	₩	95	34	35
r ₃₂	П	r ₃₂		1	r ₃₂	m		r ₃₂	r ₃₂	r ₃₂	r ₃₂				r ₃₂	m	_		32	_		\neg		Ť	ŤΓ	ΤÌ	T	_	т	Tì	T	Ť	Ť	Ť	П	m	Ħ	Т		7			
598	8	599			s16			s105	s106	s107	5101				s102									Ţ		97 1	100 1	03		104	\Box	Ţ	Ę		96				QF.			\equiv	
-	-	₩	310	08	s109	-	-	₩	-	-	+	 		-	-		-	-		-		-		⊹		₩	-	-	-	-	-	+	-	╬	Н	H	쓔		H	-	H		-
1	=	H	ras					T	1	1	1					m	_	T					\rightarrow	†	T	Ħ	T	T	т	T	\neg		┰	╈	П	m	Ħ	т		T	m	T	
Ť		mì			r ₂₁	İ		İ	Ť					Ì		T I		TÌ	ĺ					Ť	ÌΪ	ÌΠ	T			T	Tì	Ť	亡	Ť									
					r ₂₂																											\Box											
_	_	щ			r ₂₃	<u> </u>		<u> </u>	_		_			_			_	4				_		누	4	Щ	4		ш	_	_	4	_	_	Ш	Щ	Щ	_	4	4	H	_	
	-	H	s1:			-		-						-			-	-		5111 r ₄₄	3112	-		+	₩	H	-	+	+	-		+	-	+	H	H	H	H	H	+	H	\dashv	
		H	r46														7	7			r ₄₆	5113	-		+	Ħ	+	+	\top			+	-		Н	H	H	H	\vdash	Ŧ		\exists	
			r46			İ					I										r ₄₈			Ì	Ì							Ť	Ť	Ì									
					s114																				П							Ţ	Ţ								Ø		
_	_		9115 r ₅₀	-	859	-		-		-	-	-		-	-			88		50	r ₅₀		62 863 86	4	₩	H		1	\square	_		-	-	-	H	4		-	1		116	5.0	5.6
	\dashv		r ₅₂		203									+			8			52	r ₅₂		22 203 80	+	+	Ħ	- 0		T	-		+	+	+	Н	H	H	Ħ	\vdash	+	110	-	-
		m	rss			Ì		Ì				Ì		Ť							r ₅₃			Ť	T	\dagger	T		ш			Ť		T			m					\neg	
			r ₅₄																		r ₅₄			Ì																			
			rss																		r ₅₅																						
-	_	H		5118	s117			-		-	-			-	-		-	_				_		+	-	H	4	_	\blacksquare	_		+	+	-	H	4	H	\mathbb{H}	#	-			
				r ₂₅				1															-	+	+	+						+	-	+				Н					
		H		r ₂₆																				1	1	Ħ			П			7			Н	T	H	Н	rt			\dashv	
							s45																		ΪĹ							İ											
		H		-	s119			-												[+	-	1	H					1	1	-	-		H							
-		-		+	r ₃₅	-		-		-		-		-	-		-	-				-		-	₩	H	-	-	\vdash	_		+	+	-	H	4	H		+	-	\vdash	\dashv	
	-	H	51	20	r ₃₆			-							-		-	-		111		-	-	-	₩	H	+	-	+	_		+	-	-	H		H	+	+	#	H	\dashv	
r ₁₆			31	21																																			d				
r ₁₆	5	r ₁₆			r ₁₆			r ₁₆	r ₁₆	r ₁₆	r ₁₆				r ₁₆			2	16																								
			839 839	+	s38			-		-							3	37					841 842 84 841 842 84				4	0	\square			-	-	-		ŖĘ.			-			122	35 123
			839 839	+	s38 s38													37					341 342 34 341 342 34		₩	H		0	+			+	-	-	H	₽				+	\vdash	=	123
				r49																r ₄₉	r ₄₉				Ì							Ť											
-0.4	4		90		s 89												s	88					92 893 89	4			9	1 12	5												83	85	86
s84	=	$\overline{}$	313																	3111											_								$\overline{}$	=	_		

00	0121			2120			_			_	-	_		-	0120						_	120								1-1-		_
83	8131	r ₁ ;	1	8130		_	+		-		Н			_	8129			++	╁		-	128						+++	\vdash	-	┰	\vdash
85	r ₄₄			r44											r ₄₄ 3	132																
86	r46			r46											r ₄₆ r																	
87	r ₄₈			r48			-								r ₄₈ r	48 r ₄			Щ	\perp	_	\perp	\sqsubseteq					\perp	Щ.	ш	\perp	느
88	r ₅₀	8135	5	134 r ₅₀			-					-			r ₅₀ r	50 r-	-	++	₩	-	-	+						++	-	-		
90	00	560		59									s58		00	-8	s62 s6	3 564			61									136 55	5 56	57
91	r ₅₂			r ₅₂											r ₅₂ r																	
92	r ₅₃			r ₅₃	₩.					_					r ₅₃ r			\perp	Щ		_		\sqsubseteq						Щ.	Щ		느
93 94	r ₅₄			r ₅₄	-		+	\vdash	_	-	Н	-	_	-	r ₅₄ r			++	+		-	+	$\vdash\vdash$					+	\vdash	₩	+	\vdash
95	r ₅₅		r ₁₃	rss		_	+	╁	_	-	\vdash	-	_	-	r ₅₅ r s76	55 -5		++-		-	-	+							-	₩		\vdash
96			-13	_			_			_	Н	3	137	_				-		-	_	\top						1	\vdash		-	Н
97	898	899	3	16	81	.05 8106	s107	s101			s102									97 100	103		104		1	.38						
98							-	\sqcup		_	Ш	r	31	<u> </u>	Щ			+	444		_	Щ.	\square			_#		+	₩	₩.		╙
99 100	re	r ₅	r		rs	rs	rs	rs		_	rs	-			-	-	_	++	₩	+	-	+				₩			\vdash	\vdash	+-	Н
101	923						Ť																	140								
102		5141																				\blacksquare										
103	-		8142 8143	_		_	+	+	_	-	Н	-	_	-		-	-	++-		-	-	+							-	₩		\vdash
105		8144																														
106		s145		0.0								\blacksquare	-05				262	2 42			40							40	H	22 5		-
107	833	839	8	38		_	+					r ₃₇	s37				841 84	2 513	╁╟╁		40	-	146						+	34 34	4 35	30
	8149			8148								-																	147	\Box		
110				16	31	3 814	815													150			12									
111		a60 a60		59 59	-		-	+				-	a58	-	H		562 56 562 56				61 61	-						##	+	15	51 56 152	57
113		860	3	59									358				862 86	3 864			61											153
114		r ₄													r ₄₉ r	49 r ₄																
115 116		890		89									888		8111		892 89	3 894			91 15	4				₩		+	-	83 85	86	87
117			5156																													
118		839	8	38								\Box	s37				841 84	2 843			40									157 34	4 35	36
119		r ₃₃	8158			-	-					-						++		+	-	-						++	-	\vdash	+	
121			8159																		T											
122			r43												r ₄₃ s																	
123 124		+	r45	_	\vdash		-	\vdash	_	-	\vdash	\vdash	_		r ₄₅ r			+		\bot	-	+	\vdash					+	-	₩		\vdash
125		31	r ₄₇	-	\vdash	_	+		-	-	\vdash	\vdash	_	-	r ₄₇ r	47 -4		++	╁┼┼	+	-	+						+	\vdash	\vdash	+	\vdash
126			r ₅₁			\neg					П				r ₅₁ r	51 r ₅					T											
127	İ		r ₇			\equiv	Ť	ΤÏ	Ĭ	T T	ī	ΞĬ		ĺ	ΪÏ	T		Ť	TT		ΞĒ	T				TT	TT	TT	TT		一一	Ε
128		r _e		Ì			İ																									
129		890		89	\vdash	\rightarrow				_	\blacksquare		s88 s88			_	892 89 892 89				91	-			-			+	4	162 8	61 86	
131				0.9		_	-			_	Н		300			-	392 3	13 394	+		91				_				-	102 0	0.0	0.7
132		890	3	89									588				592 59				91										163	87
133 134		890	3	89		\rightarrow	-	\vdash		-	Н	\vdash	888				392 39	3 894			91	-	\vdash		\rightarrow			+	-	₩	_ _	164
135		890	- 3	r ₄₉		\rightarrow	-			-	Н		888	-	r ₄₉ r	49 -4		22 000											++	83 8	5 86	87
136		81																			91 16					—m				i i		
137															s111		892 89	3 334			91 16	00										
138 139		r ₄₂		42	r ₄	2 r ₄₂	r ₄₂	r ₄₂			r ₄₂			r ₄₂	8111			3 551			91 16	0.5							ш			
140	-		r		r ₄	2 r ₄₂	r ₄₂	r ₄₂			r ₄₂	r	30		s111		392 39													167 5	5 56	57
		r ₄₂	r	59	F4	2 r ₄₂	r ₄₂		51 852	353	r ₄₂	r			s111						61			168						167 5	5 56	57
141	s68	860	s s	59				3	51 852	953		r	30		s111		392 39						67		169					167 5	5 56	57
141	s68 r ₆	860 r ₆	s s	59 69 6	r ₆	r ₆	r ₆	r ₆	51 852	853	re	r	30		8111		392 39						67		169					167 5	5 56	57
141 142 143	## ## ## ## ## ## ## ## ## ## ## ## ##	860	s s	59		r ₆	r ₆	3	51 852	853		r	30	r ₄₂	8111		392 39	53 864					67		169					167 5		
141 142 143 144 145	s68 r ₆ r ₁₂			59 69 6	r ₆	r ₆	r ₆	r ₆	51 852	853	re		30 858	r ₄₂	8111		892 89	53 864			61		67		169							
141 142 143 144 145 146	s68 r ₆ r ₁₂	r ₆ r ₁₂ s60		59 69 6	r ₆	r ₆	r ₆	r ₆	51 952	853	re	r,	30 858	r ₄₂	8111		892 89	53 864			61		67		L69							
141 142 143 144 145 146	s68 r ₆ r ₁₂			59 69 6	r ₆	r ₆	r ₆	r ₆			re	r	30 858	r ₄₂	8111		892 89	53 864			61		67		L69							
141 142 143 144 145 146 147 148	s68 r ₆ r ₁₂		x x x x x x x x x x	59 69 6 12 59 171	r ₆	r ₆	r ₆	r ₆	51 852	953 953 953	r ₆	F 1	30 858	F42	9111		892 89	53 864			61		67		L69							
141 142 143 144 145 146 147 148 149	s68 r ₆ r ₁₂			59 69 6 12 59 171	r ₆	r ₆	r ₆	r ₆			re		30 858	F42			892 89	53 864			61		67		169							
141 142 143 144 145 146 147 148 149 150				59 69 6 12 59 171	r ₆	r ₆	r ₆	r ₆			r ₆		30 858	F42	r ₄₃ 5		862 86 862 86	53 864			61		67		169							
141 142 143 144 145 146 147 148 149 150 151				59 69 6 12 59 171	r ₆	r ₆	r ₆	r ₆			r ₆		30 858	F42	r ₄₃ 5	45 81	\$92 \$1	53 864			61		67		169							
141 142 143 144 145 146 147 148 149 150 151 152 153	## ## ## ## ## ## ## ## ## ## ## ## ##			59 69 6 12 59 171	r ₆	r ₆	r ₆	r ₆			r ₆		30 858	F42	r ₄₃ 5	45 81	\$92 \$1	53 864			61		67		1.69							
141 142 143 144 145 146 147 148 149 150 151 152	## ## ## ## ## ## ## ## ## ## ## ## ##		9 9 172 8 9 172 8 9 173 8 9 174 9 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	59 69 6 12 59 171	r ₆	r ₆	r ₆	r ₆			r ₆		30 858	F42	r ₄₃ 5	45 S1	592 50 10 10 10 10 10 10 10	53 864			61		67		169							
141 142 143 144 145 146 147 148 149 150 151 152 153 154 155	### ##################################	s60		59 69 6 12 59 171	Γ _ε Γ _ε	r ₆	Ye6 F12	r ₆ r ₁₂			r ₆		30 858	F42	r43 5 r45 r	45 S1	592 50 10 10 10 10 10 10 10	53 864			61		67		169							
141 142 143 144 145 146 147 148 149 150 151 152 153 154 155	### ##################################	s60		59 69 6 12 59 171 4 4 19 19 19	r ₆	r ₆ 2 r ₁₂	r ₆ r ₁₂ r ₄	r ₆ r ₁₂ s			r ₆		30 858	£42	r43 s	45 S1	592 50 10 10 10 10 10 10 10	53 864			61		67		169							
141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158		Fe Square Fig. 1		59 69 6 12 59 171 4 4	r ₄	r ₆ 2 r ₁₂	F ₆ F ₁₂ F ₄ F ₁₉ F ₁₄	r ₆ r ₁₂			r ₆ r ₁₂ r ₄		30 858	F42	r43 5 r45 r	45 S1	592 50 10 10 10 10 10 10 10	53 864			61		67									
141 142 143 144 145 146 147 148 150 151 152 153 154 155 156 157 158		Fe Square Fig. 1		59 69 6 12 59 1771 4 4 19 19 14 15 15 16 16 16 16 16 16 16 16 16 16 16 16 16	r ₄	r ₆ 2 F ₁₂ r ₄ 9 F ₁₉	F ₆ F ₁₂ F ₄ F ₁₉ F ₁₄	r ₆ r ₁₂ s			F ₆ F ₁₂ F ₁₄ F ₁₉		30 858	E42 E42 E44 E19 E19 E16	T43 5 T45 T47 T	45 81 47 r ₄ 51 r ₅ 7 r ₇	592 593	53 864			61		67									
141 142 143 144 145 146 147 148 149 150 151 152 153 154 158 158 158 158 159 160 161	### ##################################	Fe Square Fig. 1		59 69 6 112 559 1171 1171 1171 1171 1171 1171 1171	r _ε r ₁ r ₁ r ₁ r ₁ r ₁ r ₁ r ₁ r ₁	r ₆ 2 F ₁₂ r ₄ 9 F ₁₉	F ₆ F ₁₂ F ₄ F ₁₉ F ₁₄	r ₆ r ₁₂ s			F ₆ F ₁₂ F ₁₄ F ₁₉		30 858	E42 E44 E44 E19 E14 E15	F43 9 F46 F F45 F F47 F F51 F F76 F F43 9 F	45 81 47 r4 51 r5 7 r7 132	592 593	53 864			61				1.69							
141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161	### ##################################	Fe Square Fig. 1		59 69 6 6 12 559 1771 4 4 15 19 14 15 8 136 6	r _ε r ₁ r ₁ r ₁ r ₁ r ₁ r ₁ r ₁ r ₁	r ₆ 2 F ₁₂ r ₄ 9 F ₁₉	F ₆ F ₁₂ F ₄ F ₁₉ F ₁₄	r ₆ r ₁₂ s			F ₆ F ₁₂ F ₁₄ F ₁₉		30 858	E42 E42 E44 E44 E19	T43 9 7 447 T 51	45 81 47 r ₄ 51 r ₅ 7 r ₇ 132	952 952 95	53 864			61	176			1.69							
141 142 143 144 145 146 147 148 149 150 151 152 155 154 155 157 168 159 160 161	### ### #### #########################	Fe Square Fig. 1		59 69 69 6 6 112 2 1559 1771 4 4 4 15 15 15 15 15 15 15 15 15 15 15 15 15	r _ε r ₁ r ₁ r ₁ r ₁ r ₁ r ₁ r ₁ r ₁	r ₆ 2 F ₁₂ r ₄ 9 F ₁₉	F ₆ F ₁₂ F ₄ F ₁₉ F ₁₄	r ₆ r ₁₂ s			F ₆ F ₁₂ F ₁₄ F ₁₉		30 858	E42 E44 E44 E19 E14 E15		45 81 47 r4 51 r5 7 r7 132	992 93 93	53 864			61				65							
141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 164 165	### ##################################	Fe Square Fig. 1		59 59 68 68 68 68 68 69 122 559 171 17	r _ε r ₁ r ₁ r ₁ r ₁ r ₁ r ₁ r ₁ r ₁	r ₆ 2 F ₁₂ r ₄ 9 F ₁₉	F ₆ F ₁₂ F ₄ F ₁₉ F ₁₄	r ₆ r ₁₂ s			F ₆ F ₁₂ F ₁₄ F ₁₉		30 858	E42 E42 E44 E44 E44 E45	F40 5 F46 F F47 F F43 5 F47 F F43 5 F47 F F43 5 F47 F F44 5 F F44 5 F F44 7 F F44 7 F F44 7 F	45 81 47 r ₄ 51 r ₅ 7 r ₇ 132 45 81 47 r ₄	992 93 93 93 94 94 94 94 94 94 94 94 94 94 94 94 94	53 864			61				650							
141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 164 165	### ##################################	G		59 69 69 69 69 60 60 60 60 60 60 60 60 60 60 60 60 60	r _ε r ₁ r ₁ r ₁ r ₁ r ₁ r ₁ r ₁ r ₁	r ₆ 2 F ₁₂ r ₄ 9 F ₁₉	F ₆ F ₁₂ F ₄ F ₁₉ F ₁₄	r ₆ r ₁₂ s			F ₆ F ₁₂ F ₁₄ F ₁₉		30 858	E42 E4 E4 F19 E14 F15	F43 8 F45 F47 F51 F43 F45 F47 F51 F45 F47 F51 F45 F47 F45 F47 F51 F45 F47 F45 F47 F45 F47 F51 F51 F51 F51 F51 F47 F51 F51 F51 F51 F51 F51 F51 F51 F51 F51	45 81 74 51 74 51 75 77 77 77 74 74 74 74 74 75 75 75 75 75 75 75 75 75 75 75 75 75	992 93 93 93 94 94 94 94 94 94 94 94 94 94 94 94 94	53 864			61											
141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 160 161 162 163 164 165 166 166 167	### ##################################	F6 F12 F44 F44 F44 F44 F44 F44 F44 F44 F44 F4		59 69 69 6 6 112 59 171 4 4 115	r _ε r ₁ r ₁ r ₁ r ₁ r ₁ r ₁ r ₁ r ₁	r ₆ 2 F ₁₂ r ₄ 9 F ₁₉	F ₆ F ₁₂ F ₄ F ₁₉ F ₁₄	r ₆ r ₁₂ s			F ₆ F ₁₂ F ₁₄ F ₁₉		30 858	E42 E4 E4 F19 E14 F15	F40 5 F46 F F47 F F48 5 F47 F F48 5 F47 F F48 7 F F48 7 F F48 7 F F49	45 81 74 51 74 51 75 77 77 77 74 74 74 74 74 75 75 75 75 75 75 75 75 75 75 75 75 75	992 93 93 93 94 94 94 94 94 94 94 94 94 94 94 94 94	53 864			61				169							
141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 160 161 162 163 164	# # # # # # # # # # # # # # # # # # #	E6 F12 F61 F		59 59 68 68 68 68 68 69 122 559 171 17	r _ε r ₁ r ₁ r ₁ r ₁ r ₁ r ₁ r ₁ r ₁	r ₆ 2 F ₁₂ r ₄ 9 F ₁₉	F ₆ F ₁₂ F ₄ F ₁₉ F ₁₄	r ₆ r ₁₂ s			F ₆ F ₁₂ F ₁₄ F ₁₉		30 858	E42 E44 E45 E45 E15	F43 8 F45 F47 F51 F43 F45 F47 F51 F45 F47 F51 F45 F47 F45 F47 F51 F45 F47 F45 F47 F45 F47 F51 F51 F51 F51 F51 F47 F51 F51 F51 F51 F51 F51 F51 F51 F51 F51	45 81 74 51 74 51 75 77 77 77 74 74 74 74 74 75 75 75 75 75 75 75 75 75 75 75 75 75	992 93 93 93 93 93 93 93 93 93 93 93 93 93	53 864			61				.69							

71				s182																															
72 r	16	r ₁₆			r ₁₆				r ₁₆	r ₁₆	r ₁₆	r16	r ₁₆																	ш					
.73					s18	33																													
.74				r7			П										r ₇	r ₇	r7																
.75 s	187				s18	88	\neg									186		$\neg \vdash$		$\neg \neg$					185			184						\Box	
76		\Box	\neg	r ₉	$\neg \vdash$		$\neg r$				\Box		\neg					$\neg \vdash$					$\neg \Box$		$\neg \Box$				П		$\Box\Box$		\Box	\Box	
.77 r	r ₇	m				r	T							m	T		r ₇	r ₇	r ₇				T						m					ŤΪ	
78		т			s1	5	寸		s105	s106	s107		$\overline{}$					\neg				\neg	\top	189 103	104		T		m				\Box	T	ΠÌ
79		T			190		T							i	T			\neg	Ť			Tit	T						m	ĦΠ				Ħ	ΠÌ
80			839		838	3									1	37				841 842	843			40									193	34	35 3
81					192																														
82					193				9																										
83 9	149					31	148																									194			
84				3195																															
85				r ₂₇																															
86					s19	96																													
87				r ₂₉																															
88							3	197																											
89 r	4	r ₄			r ₄		\neg		r ₄	r ₄	r ₄	r ₄	r ₄					Т				П							П					П	
90 r	19	r19			r19		\neg		r ₁₉	r ₁₉	r ₁₉	r ₁₉	r19																						
91		т			198	T	十				$\overline{}$		\neg	i			37	6	\vdash	\neg			ī		$\neg \neg$				m		$\neg \neg$		\Box	m	
92 r	r ₁₄	r ₁₄			r ₁₄		\neg		r ₁₄	r ₁₄	r ₁₄	r ₁₄	r ₁₄																	$\sqcap \sqcap$					
93 r	15	r ₁₅			r ₁₅		\neg		r ₁₅	r ₁₅	r ₁₅	r ₁₅	r ₁₅																						
94				r40			Ť							í	Tì			Ť		i	ΤĖ	Tì				=			m					m	
95		т			_	_	一	_			-	_	_	5199	T			_	1			TH	_						m				$\overline{}$	T	m
96		m		r ₂₈	\neg	_				-		-	i					7	1		m					\neg			m				$\overline{}$	m	
97			s60	-	859	-	十				-	_	_			s58		_	+	862 863	364		_	61				_	m	\vdash	\vdash		200	55	56 5
98 8	187				818		十		,				Ť	i		186		\neg							185			201							
99 9	98	899	i		s1		一		s105	s106	s107	s101	s102	m				ì			T		97	100 103	104		m		202	ТΠ			\vdash	i	ΠÌ
00				r ₁₃	T	T	Ť							ΠÌ			51	11			T						П							T	
01		П		3203			Ť															TH			\neg		T		m	ТΠ			m	m	
02					$\overline{}$		十								204			\neg							一		T		m						
03							Ť							s205																					
04 r	24	r24	Î		r ₂₄				r ₂₄	r24	r24	r24	r24				r ₂₄																	í	
05 B		899			81		7				8107		8102								\vdash	m	97	100 103	104				206				\Box	Ì	
06		П	T				一				Ì				207			ì					T				m		m	ΤП				m	
07 r		r ₂₄			r ₂₄	T T	$\overline{}$		r ₂₄	r ₂₄	ro.	r ₂₄	r24	ī				\neg									T								

Como puede observarse en la tabla[2], esta gramática es adecuada para este tipo de analizador sintáctico, puesto que no se produce ningún tipo de conflicto.

4 Diseño Analizador Semántico

```
Acción semántica previa a empezar a funcionar el procesador:
\{ TS_g = crear_TS() \}
 TS\_actual = TS\_g
 desp = 0
 zona decl = false
B \to D \{ \}
D \to F D \{\}
D \to G D \{\}
D \rightarrow lambda \{\}
G \rightarrow if (E) S \{ if E.tipo != "lógico" \}
                 then error(1) }
G \to S \{ \}
S \to H ; \{\}
H \rightarrow id (I)  { if busca_tipo_TS(id.pos) != "función"
                 then error(15)
             else if longitud(I.tipo) != busca num params TS(id.pos)
                 then error(2)
             else if busca_tipo_TS(id.pos) != I.tipo
                 then error(3)
             else
                 H.tipo = busca tipo devuelto TS(id.pos) }
I \rightarrow E J \{ \text{ if longitud}(J.tipo) == 0 \}
             then I.tipo = E.tipo
          else
             I.tipo = E.tipo x J.tipo }
J \rightarrow E J1  if longitud(J1.tipo) == 0
             then J.tipo = E.tipo
            else
             J.tipo = E.tipo x J1.tipo 
J \rightarrow lambda \{\}
I \rightarrow lambda \{ I.tipo = void \}
S \to K ; \{\}
K \rightarrow id = E \{ if busca\_tipo\_TS(id.pos) != None \}
                 añade tipo TS(id.pos, "entero")
                 a\tilde{n}ade\_desp\_TS(id.pos, desp)
                 desp+=2
            if busca tipo TS(id.pos)!= E.tipo
                 then error(10) }
```

```
S \rightarrow alert (E); { if E.tipo != "cadena" && E.tipo != "entero"
                    then error(4) }
S \rightarrow input (id); { if busca_tipo_TS(id.pos) != None
                       añade tipo TS(id.pos, "entero")
                       añade desp TS(id.pos, desp)
                       desp+=2
                  if busca_tipo_TS(id.pos) != "cadena"
                     && busca_tipo_TS(id.pos) != "entero"
                       then error(5) }
S \rightarrow \text{return L}; { if zona function != true
                 then error(7)
               else if L.tipo!= tipo return
                 then error(9) }
L \rightarrow E \{ L.tipo = E.tipo \}
L \rightarrow lambda \{ L.tipo = void \}
G \rightarrow let M T id; { añadir_tipo_TS(id.pos, T.tipo)
                 añadir desp TS(id.pos, desp)
                 desp += T.ancho
                 zona declaración = false
M \rightarrow lambda \{ zona declaracion = true \}
T \rightarrow number \{ T.tipo = "entero" \}
            T.ancho = 2 // size of("entero")
T \rightarrow boolean \{ T.tipo = "lógico" \}
            T.ancho = 2 \} // \ size\_of("l\'ogico")
T \rightarrow string \{ T.tipo = "cadena" \}
             T.ancho = 128 // size of("cadena")
G \rightarrow for (N; E; O) \{C\} \{if E.tipo! = "lógico"\}
                              then error(6) }
N \to K \{ \}
N \rightarrow lambda \{\}
O \to K \{\}
O \rightarrow -- id \{ \text{ if busca tipo } TS(id.pos) \} = None
              añade_tipo_TS(id.pos, "entero")
              añade desp TS(id.pos, desp)
              desp+=2
           if busca_tipo_TS(id.pos) != "entero"
              then error(11) }
O \rightarrow lambda \{\}
C \to G C \{\}
C \rightarrow lambda \{\}
```

```
F \rightarrow F1 F2 F3 \{ destruye TS (TS 1) \}
             zona function = false
             TS\_actual = TS\_g
             desp = desp g
             tipo return = NULL }
F1 \rightarrow function P Q id \{ TS_l = crear_TS() \}
                    TS\_actual = TS\_l
                    desp_g = desp
                    desp = 0
                    pos id fun = id.pos
                    zona function = true
                    tipo return = Q.tipo
                    añadir_tipo_devuelto_TS(id.pos, tipo_return)
                    añadir tipo TS(id.pos, "función")
                    inserta et TS (id.pos, nueva et ()) }
P \rightarrow lambda \{\}
Q \rightarrow T \{ zona\_declaracion = true \}
       Q.tipo = T.tipo }
Q \rightarrow lambda \{ zona declaracion = true \}
            Q.tipo = void 
F2 \rightarrow (A) { añadir param TS(pos id fun, A.tipo, longitud(A.tipo))
                     zona declaración = false
A \rightarrow T \text{ id } AA \{ \text{ añadir tipo } TS(\text{id.pos}, T.tipo) \}
            añadir_desp_TS(id.pos, desp)
            desp += T.ancho
            if longitud(AA.tipo) == 0
               then A.tipo = T.tipo
            else
               A.tipo = T.tipo \times AA.tipo 
A \rightarrow lambda \{ A.tipo = void \}
AA \rightarrow T id AA1  { añadir tipo TS(id.pos, T.tipo)
                 añadir desp TS(id.pos, desp)
                 desp += T.ancho
                 if longitud(AA1.tipo) == 0
                    then AA.tipo = T.tipo
                 else
                    AA.tipo = T.tipo \times AA1.tipo }
AA \rightarrow lambda \{\}
F3 \rightarrow \{C\} \{\}
E \rightarrow E1 \&\& R  { if E1.tipo != "lógico" || R.tipo != "lógico"
             then error(12)
            else
             E.tipo = "lógico" }
E \rightarrow R \{E.tipo = R.tipo \}
```

```
R \rightarrow R1 == U { if R1.tipo != "entero" || U.tipo != "entero"
             then error(13)
             E.tipo = "lógico" 
R \rightarrow U \{ R.tipo = U.tipo \}
U \rightarrow U1 - V { if U1.tipo != "entero" || V.tipo != "entero"
             then error(14)
           else
             U.tipo = "entero" }
U \rightarrow V \{ U.tipo = V.tipo \}
V \rightarrow -- id \{ \text{ if busca tipo } TS(id.pos) != \text{None} \}
             añade tipo TS(id.pos, "entero")
             añade_desp_TS(id.pos, desp)
             desp+=2
          if busca_tipo_ TS(id.pos) != "entero"
             then error(15)
          else
             V.tipo = "entero" }
V \rightarrow id { if busca tipo TS(id.pos) != None
           añade_tipo_TS(id.pos, "entero")
           añade desp TS(id.pos, desp)
           desp+=2
        V.tipo = busca tipo TS(id.pos)
V \rightarrow (E) \{ V.tipo = E.tipo \}
V \rightarrow H \{ V.tipo = H.tipo \}
V \rightarrow \text{ent} \{ V.\text{tipo} = \text{"entero"} \}
V \rightarrow cad \{ V.tipo = "cadena" \}
V \rightarrow \log \{ V.tipo = "lógico" \}
error\_code\_dict = \{
   1: "La condición debe ser un lógico",
   2: "El número de parámetros introducidos no son los esperados, \
      deberían ser {busca num params TS(id.pos)}",
   3: "El tipo de los parámetros no es el esperado, \
      se esperaban {busca_tipo_params_TS(id.pos)}",
   4: "La expresión introducida no es una cadena o un entero",
   5: "La variable introducida no es de tipo cadena o entero",
   6: "La condición debe ser un lógico",
   7: "No puede haber una sentencia return fuera de una función",
   8: "No se permite la definición de funciones anidadas",
   9: "El tipo de retorno no corresponde con el tipo de retorno de la función, \
      se esperaba {tipo return}",
   10: "El tipo de la variable a asignar no corresponde con el tipo asignado",
   11: "El operador especial "--" solo trabaja con tipos de datos enteros",
   12: "El operador lógico "&&" solo trabaja con tipos de datos lógicos",
   13: "El operador de relación "==" solo trabaja con tipos de datos enteros",
   14: "El operador aritmético "-" solo trabaja con tipos de datos enteros",
   15: "La variable no se puede invocar como una función"
}
```

```
function error(error_code):
    res = ""
    if error_code <= 6:
        res = "ErrorDeAtributo: "
    elif error_code >= 9:
        res = "ErrorDeTipado: "
    else
        res = "NoImplementado: "
    res = res + error_code_dict.get(error_code, default="Código no válido")
    print(res)
```

5 | Diseño Tabla de Símbolos

La TS se compone de una lista de tablas, una de ellas es global y se crea al empezar, mientras que el resto son locales y se van creando a medida que avanza la compilación. Están ordenadas en orden de creación. Cada tabla tiene un flag que indica si la tabla existe (esta activa) o se ha eliminado, pero realmente no las eliminamos para posteriormente imprimirlas.

A su vez cada tabla contiene una lista de diccionarios, cada diccionario simboliza una entrada en la tabla de símbolos. Un diccionario es un hashmap que tiene como claves la palabra "lexema" y los tipos de atributos que le corresponda (por ejemplo "Tipo" ó "Despl"), como valores tiene el valor del lexema y de sus atributos en si mismos (por ejemplo "main" ó "entero").

6 Anexo

6.1 Casos de prueba correctos

6.1.1 Prueba correcta 1

Programa introducido:

```
let string texto;
let string textoAux;
textoAux = texto;
alert
(textoAux);
```

Tokens:

```
<LET , >
<\!\!\text{STRING}\ ,>
<ID , 0>
<PUNTOYCOMA , >
{<}\mathrm{LET} , {>}
<STRING , >
<ID, 1>
<PUNTOYCOMA , >
<ID , 1>
<OPASIG , >
<ID , 0>
<PUNTOYCOMA, >
<ALERT , >
<ABPAREN , >
<ID, 1>
<CEPAREN ,>
<PUNTOYCOMA , >
```

Tabla de símbolos:

CONTENIDO DE LA TABLA 0:

```
* LEXEMA: 'texto'
ATRIBUTOS:
+ Tipo: 'cadena'
+ Despl: '0'

* LEXEMA: 'textoAux'
ATRIBUTOS:
+ Tipo: 'cadena'
+ Despl: '64'
```

VASt:

```
Ė-B (1)
 Ġ-G (20)
      --let
⊟-M (21)
       --lambda
      □ T (24)
       string
       ...id
....;
    □-D (3)
      ☐ G (20)
        ...let
⊟.·M (21)
         lambda
         T (24)
         string
         ...id
      ...D (3)
         Ġ (6)
          .
-S (13)
             Ė-K (14)
               ...id
               E (45)
                .:⊢R (47)
                   .:.U (49)
                     i...id
         □-D (3)
           Ġ (6)
             Ė-S (15)
                alert
                ---(
               E (45)
                Ė-R (47)
                  ...id
           □ D (4)
             lambda
```

6.1.2 Prueba correcta 2

Programa introducido:

```
function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number Factorial (number n)

function number function number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number number
```

Tokens:

```
<FUNCTION, >
<NUMBER , >
<ID , 0>
<ABPAREN, >
<NUMBER , >
<ID , 0>
<CEPAREN , >
<ABLLAVE , >
<IF ,>
<ABPAREN , >
<ID , 0>
<OPREL, >
<CTEENTERA , 0>
<CEPAREN ,>
<RETURN , >
<CTEENTERA, 1>
<PUNTOYCOMA, >
<\!\!\mathrm{RETURN}\ ,>
<ID , 0>
<OPARIT , >
<ID , 0>
<ABPAREN , >
<ID , 0>
<OPARIT , >
<CTEENTERA , 1>
<CEPAREN ,>
<PUNTOYCOMA, >
<CELLAVE, >
```

Tabla de símbolos:

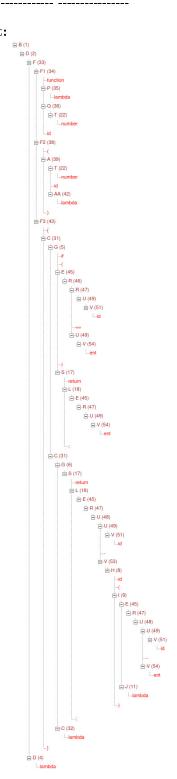
CONTENIDO DE LA TABLA 0:

```
* LEXEMA: 'Factorial'
ATRIBUTOS:
+ Tipo: 'funcion'
+ TipoRetorno: 'entero'
+ EtiqFuncion: 'Et_Fun_0'
+ numParam: '1'
+ TipoParam1: 'entero'
```

CONTENIDO DE LA TABLA 1:

LEXEMA: 'n' ATRIBUTOS: + Tipo: 'entero' + Despl: '0'

VASt:



6.1.3 Prueba correcta 3

Programa introducido:

```
1 let boolean z;
2 for (b=1;z; )
3 {
4 alert (88);
5 }
```

Tokens:

```
{<}\mathrm{LET} , {>}
<BOOLEAN, >
<ID , 0>
<PUNTOYCOMA , >
<FOR , >
<\!\!\mathrm{ABPAREN}\ ,>
<ID , 1>
<\! \mathrm{OPASIG} \ ,>
<CTEENTERA , 1>
<PUNTOYCOMA , >
<ID , 0>
<PUNTOYCOMA, >
<CEPAREN , >
<ABLLAVE , >
<ALERT , >
<ABPAREN, >
<\!CTEENTERA , 88>
<\! {\rm CEPAREN} \ , >
<PUNTOYCOMA, >
<CELLAVE, >
```

Tabla de símbolos:

CONTENIDO DE LA TABLA 0:

```
* LEXEMA: 'z'
ATRIBUTOS:
+ Tipo: 'logico'
+ Despl: '0'

* LEXEMA: 'b'
ATRIBUTOS:
+ Tipo: 'entero'
+ Despl: '1'
```

VASt:

```
Ė-B (1)
Ė ·D (3)
    Ġ (20)
      ...let
⊟-M (21)
       lambda
       □ T (23)
        boolean
       ...id
     ⊡..D (3)
       □ G (25)
        ...for
...(
....( 26)
           Ė K (14)
            ...id
...=
             <u>-</u>E (45)
              .÷.R (47)
               Ė V (54)
                       ent
         --;
---E (45)
           Ė-R (47)
            ...U (49)
             Ė.·V (51)
               i...id
          —O (30)
          --lambda
         --)
--{
⊡--C (31)
           □ G (6)
             Ė-S (15)
               ...alert
...(
                E (45)
                  Ė-R (47)
                   Ė ·U (49)
                     ....V (54)
                          ent
            □-C (32)
            lambda
       □ D (4)
         lambda
```

6.1.4 Prueba correcta 4

Programa introducido:

```
1
   /* prueba
2
                correcta */
     let boolean b; let number x;
   input (z);
   alert (z);
   x=z;
6
   alert (z-1);
   b=b&&b;if (b)
   x =
9
10
     x - 6
       - z
11
        - 1
12
        - (2
13
14
        - у
        - 6);
```

Tokens:

```
{<}\mathrm{LET} , {>}
<BOOLEAN , >
<ID, 0>
<PUNTOYCOMA , >
<LET , >
<NUMBER , >
<ID , 1>
<PUNTOYCOMA , >
<INPUT , >
<ABPAREN, >
<ID , 2>
<CEPAREN , >
<PUNTOYCOMA , >
<ALERT , >
<ABPAREN , >
{<}\mathrm{ID} , {2}{>}
<\! {\rm CEPAREN} \ , >
<PUNTOYCOMA , >
<ID , 1>
<OPASIG , >
<ID , 2>
<PUNTOYCOMA, >
<ALERT , >
<ABPAREN , >
<ID , 2>
<OPARIT, >
<\!CTEENTERA , 1>
<CEPAREN ,>
<PUNTOYCOMA, >
<ID , 0>
<OPASIG , >
<ID , 0>
<OPLOG , >
```

```
<ID , 0>
  <PUNTOYCOMA, >
  <IF , >
  <ABPAREN , >
  <ID , 0>
  <CEPAREN , >
  <ID , 1>
  <OPASIG , >
  {<}\mathrm{ID} , 1>
  <OPARIT, >
  <\!CTEENTERA , 6>
  <OPARIT, >
  <ID , 2>
  <OPARIT , >
  <CTEENTERA , 1>
  <OPARIT , >
  <ABPAREN , >
  <CTEENTERA , 2>
  <OPARIT, >
  {<}\mathrm{ID} , {3}{>}
  <\! \mathrm{OPARIT} \ , >
  <CTEENTERA, 6>
  <CEPAREN, >
  <PUNTOYCOMA , >
Tabla de símbolos:
  CONTENIDO DE LA TABLA 0:
        LEXEMA: 'b'
       ATRIBUTOS:
       + Tipo : 'logico'
       + Despl: '0'
      -----
        LEXEMA: 'x'
       ATRIBUTOS:
       + Tipo: 'entero'
       + Despl: '1'
        LEXEMA: 'z'
       ATRIBUTOS:
       + Tipo: 'entero'
       + Despl: '2'
        LEXEMA: 'y'
       ATRIBUTOS:
       + Tipo : 'entero'
```

+ Despl: '3'

VASt:



6.1.5 Prueba correcta 5

Programa introducido:

```
function FuncionSentencia (number b, boolean z)
2
       for (b=0;true; --b )
3
4
       {
       alert ("hola");
5
       }
6
7
   function Funcion (number x, boolean b)
8
9
       if (x == 0) FuncionSentencia(x,b);
10
11
       alert
            (cadena); return;
12
  }
13
```

```
Tokens:
  <FUNCTION, >
  <ID , 0>
  <ABPAREN , >
  <NUMBER , >
  <ID, 0>
  <COMA , >
  <BOOLEAN, >
  <ID , 1>
  <CEPAREN , >
  < ABLLAVE, >
  <FOR , >
  <ABPAREN , >
  <ID , 0>
  <OPASIG, >
  <CTEENTERA , 0>
  <PUNTOYCOMA , >
  <CTELOGICA , 1>
  <PUNTOYCOMA, >
  <OPESP , >
  <ID , 0>
  < CEPAREN, >
  <ABLLAVE , >
  <ALERT , >
  <ABPAREN , >
  <\! CADENA , "hola">
  <\! {\rm CEPAREN} \ , >
  <PUNTOYCOMA, >
  <\! \mathrm{CELLAVE} \ , >
  <CELLAVE, >
  <FUNCTION, >
  <ID , 1>
  <ABPAREN , >
  <NUMBER, >
  <\!\mathrm{ID} , 0\!>
  <COMA , >
  <BOOLEAN, >
```

```
<ID , 1>
  <CEPAREN , >
  <ABLLAVE, >
  <IF , >
  <ABPAREN , >
  <ID , 0>
  <OPREL , >
  <CTEENTERA , 0>
  <CEPAREN ,>
  <ID , 0>
  <ABPAREN , >
  <ID , 0>
  <COMA , >
  <ID , 1>
  <CEPAREN,>
  <PUNTOYCOMA , >
  <ALERT , >
  <ABPAREN , >
  <ID, 2>
  <CEPAREN,>
  <PUNTOYCOMA, >
  <RETURN , >
  <PUNTOYCOMA, >
  <CELLAVE , >
Tabla de símbolos:
  CONTENIDO DE LA TABLA 0:
        LEXEMA: 'FuncionSentencia'
       ATRIBUTOS:
       + Tipo: 'funcion'
       + TipoRetorno : 'void'
       + EtiqFuncion: 'Et Fun 0'
       + numParam: '2'
       + TipoParam1 : 'entero'
       + TipoParam2: 'logico'
        LEXEMA: 'Funcion'
       ATRIBUTOS:
       + Tipo: 'funcion'
       + TipoRetorno: 'void'
       + EtiqFuncion: 'Et Fun 1'
       + numParam : '2'
       + TipoParam1 : 'entero'
       + TipoParam2: 'logico'
        LEXEMA: 'cadena'
       ATRIBUTOS:
       + Tipo: 'entero'
       + Despl: '0'
```

CONTENIDO DE LA TABLA 1:

* LEXEMA: 'b' ATRIBUTOS:

 $+ \ Tipo: 'entero' \\ + \ Despl: '0'$

* LEXEMA: 'z'
ATRIBUTOS:
+ Tipo: 'logico'

+ Despl : '1'

CONTENIDO DE LA TABLA 2:

* LEXEMA: 'x'
ATRIBUTOS:

 $+ \ Tipo: 'entero' \\ + \ Despl: '0'$

* LEXEMA : 'b'

ATRIBUTOS:

+ Tipo : 'logico' + Despl : '1'

VASt:



6.2 Casos de prueba erróneos

6.2.1 Prueba errónea 1

Programa introducido:

```
1 let number id;
2 id(2);
```

Mensaje de error:

Error en la linea 2:

La variable no se puede invocar como una función, con argumentos

6.2.2 Prueba errónea 2

Programa introducido:

```
1 let boolean id;
2 res = --id;
```

Mensaje de error:

Error en la linea 2:

El operador especial '--' solo trabaja con tipos de datos enteros

6.2.3 Prueba errónea 3

Programa introducido:

```
function Funcion (number a, boolean b, string c){}
funcion(a,a,a);
```

Mensaje de error:

Error en la linea 2:

El tipo de los parámetros no es el esperado, se esperaban "'entero', 'logico', 'cadena'"

6.2.4 Prueba errónea 4

Programa introducido:

```
let string texto;
function pideTexto ()
{
   alert ("Introduce un texto");
   input (texto);
}

return;
```

Mensaje de error:

Error en la linea 8:

No puede haber una sentencia return fuera de una función

6.2.5 Prueba errónea 5

Programa introducido:

```
1 let boolean bool;
2 alert(bool);
```

Mensaje de error:

Error en la linea 2:

La expresión introducida no es una cadena o un entero

7 Referencias

$1. \quad Documentaci\'on\ libreria\ SLY$

https://sly.readthedocs.io/en/latest/

2. Generador de tabla LR(1)

http://jsmachines.sourceforge.net/machines/lr1.html