Analizador Sintactico

Gramática:

1-<PROGRAM> -> DECLARE_LIBRARY PROGRAM2 DECLARA_CLASS

2-<DECLARE_LIBRAR> -> include id . lib ; <DECLARE_LIBRAR>
3-<DECLARE_LIBRAR> -> vacío

4-<PROGRAM2>-> DECLARAVAR PROGRAM2
5-<PROGRAM2>-> DECLARACONST PROGRAM2
6-<PROGRAM2>-> DECLARA_FUNCTION PROGRAM2
7-<PROGRAM2>-> vacío

8-<DECLARA_CLASS> -> class ESTATUTOS endclass

9-<DECLARAVAR> -> def DECLARAVAR2 of TIPO_DE_DATO ;
10-<DECLARAVAR> -> vacío
11-<DECLARAVAR2> -> id DECLARAVAR2A
12- <DECLARAVAR2A> -> vacío
13-<DECLARAVAR2A> -> , id DECLARAVAR2A

14-<DECLARACONST> -> const id = VALOR_CONSTANTE ;
15-<DECLARACONST> -> vacío

16-<DECLARA_FUNCTION>-> function id = TIPO_DE_DATO (PARAMETROS) DECLARAVAR ESTATUTOS end_function

17-<ESTATUTOS> -> EST_ID ; <ESTATUTOS>
18-<ESTATUTOS> -> EST_RETURN ; <ESTATUTOS>
19-<ESTATUTOS> -> EST_WRITE ; <ESTATUTOS>
20-<ESTATUTOS> -> EST_READ ; <ESTATUTOS>
21-<ESTATUTOS> -> PREINCUNARIO ; <ESTATUTOS>
22-<ESTATUTOS> -> EST_IF <ESTATUTOS>
23-<ESTATUTOS> -> EST_WHILE <ESTATUTOS>
24-<ESTATUTOS> -> EST_DO <ESTATUTOS>
25-<ESTATUTOS> -> EST_FOR <ESTATUTOS>
26-<ESTATUTOS> -> vacío

27-<EST_ID> -> id EST_ID'
28-<EST_ID'> -> = EXPR
29-<EST_ID'> -> POSTINCUNARIO'

30-<TIPO_DE_DATO> -> int
31-<TIPO_DE_DATO> -> float
32-<TIPO_DE_DATO> -> char
33-<TIPO_DE_DATO> -> string
34-<TIPO_DE_DATO> -> bool
35-<TIPO_DE_DATO> -> void
36-<VALOR_CONSTANTE> -> cteentera
37-<VALOR_CONSTANTE> -> ctereal

Analizador Sintactico

38-<VALOR_CONSTANTE> -> ctenotacion
39-<VALOR_CONSTANTE> -> ctecaracter
40-<VALOR_CONSTANTE> -> ctedtring

41-<PARAMETROS> -> PARAMETROS2 = TIPO_DE_DATO , PARAMETROS
42-<PARAMETROS> -> PARAMETROS2 = TIPO_DE_DATO
43-<PARAMETROS> -> vacío
44-<PARAMETROS2> -> id PARAMETROS2A
45-<PARAMETROS2> -> vacío
46-<PARAMETROS2A> -> , id PARAMETROS2A

47-<EST_RETURN> -> return EXPR

48-<EST_WRITE> -> write (EXPR EST_WRITE2)
49-<EST_WRITE2> -> , EXPR EST_WRITE2
50-<EST_WRITE2> -> vacío

51-<EST_READ> -> read (id EST_READ2)
52-<EST_READ2> -> , id EST_READ2
53-<EST_READ2> -> vacío

54-<POSTINCUNARIO> -> POSTINCUNARIO'
55-<POSTINCUNARIO'> -> ++
56-<POSTINCUNARIO'> -> - -

57-<PREINCUNARIO> -> ++ id
58-<PREINCUNARIO> -> - - id

59-<EST_IF> -> if (EXPR) ESTATUTOS EST IF2 EST IF3 endif
60-<EST IF2> -> elseif (EXPR) ESTATUTOS EST IF2
61-<EST IF2> -> vacío
62-<EST IF3> -> else ESTATUTOS
63-<EST IF3> -> vacío

64-<EST_WHILE> -> while (EXPR) ESTATUTOS endwhile

65-<EST_DO> -> do ESTATUTOS dowhile (EXPR) enddo

66-<EST_FOR> -> for id (EXPR to EXPR) ESTATUTOS endfor

67-<EXPR> -> EXPR2 EXPR'
68-<EXPR'> -> || EXPR2 EXPR'
69-<EXPR'> -> vacío

70-<EXPR2> -> EXPR3 EXPR2'
71-<EXPR2'> -> && EXPR3 EXPR2'
72-<EXPR2'> -> vacío

73-<EXPR3> -> ! EXPR4

Analizador Sintactico

74-<EXPR3> -> EXPR4

75-<EXPR4>-> EXPR5 EXPR4_2
76-<EXPR4_2>-> OPREL EXPR5
77-<EXPR4_2>-> vacío

78-<EXPR5> -> TERM EXPR5'
79-<EXPR5'> -> + EXPR5
80-<EXPR5'> -> - EXPR5
81-<EXPR5'> -> vacío

82-<OPREL>-> ==
83-<OPREL>-> !=
84-<OPREL>-> <
85-<OPREL>-> <=
86-<OPREL>-> >
87-<OPREL>-> >=

88-<TERM>-> FACT TERM'
89-<TERM'>-> * FACT TERM'
90-<TERM'>-> / FACT TERM'
91-<TERM'>-> % FACT TERM'
92-<TERM'>-> ** FACT TERM'
93-<TERM'>-> vacío

94-<FACT>-> id LLAMADA_F
95-<FACT>-> VALOR_CONSTANTE
96-<FACT>-> (EXPR)

97-<LLAMADA_F>-> (LLAMADA_F2)
98-<LLAMADA_F>-> vacío
99-<LLAMADA_F2>-> id LLAMADA_F2A
100-<LLAMADA_F2>-> vacío
101-<LLAMADA_F2A>-> , id LLAMADA_F2A
102-<LLAMADA_F2A>-> vacío

Analizador Sintactico

Factorizaciones:

<POSTINCUNARIO> -> id ++
<POSTINCUNARIO> -> id - -
<POSTINCUNARIO> -> id POSTINCUNARIO'
<POSTINCUNARIO'>-> ++
<POSTINCUNARIO'>-> - -

—----------

<EXPR> -> EXPR2 || EXPR
<EXPR> -> EXPR2

<EXPR> -> EXPR2 || EXPR'
<EXPR> -> EXPR
<EXPR> -> vacío

—----------

<EXPR2> -> EXPR3 && EXPR2
<EXPR2> -> EXPR3 &&

<EXPR2> -> EXPR3 && EXPR3'
<EXPR2> -> EXPR2
<EXPR2> -> vacío

—----------

<EXPR5>-> TERM
<EXPR5>-> TERM + EXPR5
<EXPR5>-> TERM - EXPR5

<EXPR5> -> TERM EXPR5'
<EXPR5'> -> + EXPR5
<EXPR5'> -> - EXPR5
<EXPR5'> -> vacío

—----------

<TERM>-> FACT
<TERM>-> FACT * TERM
<TERM>-> FACT / TERM
<TERM>-> FACT % TERM
<TERM>-> FACT ** TERM

<TERM>-> FACT TERM'
<TERM'>-> * FACT TERM'
<TERM'>-> / FACT TERM'
<TERM'>-> % FACT TERM'
<TERM'>-> ** FACT TERM'
<TERM'>-> vacío

Analizador Sintactico

FIRST:

First (PROGRAM) = { } = **{include, def, const, function, class }**

FIRST(<DECLARE_LIBRARY>) = **{ include, ε }**

FIRST(<PROGRAM2>) = **{ def, const, function, ε }**

FIRST(<DECLARAVAR>) = **{ def, ε**
FIRST(<DECLARAVAR2>) **= { id }**
FIRST(<DECLARAVAR2A>) = **{ ε , }**

FIRST(<DECLARACONST>) = **{ const, ε }**

FIRST(<DECLARA_FUNCTION>) = **{ function }**

FIRST(<DECLARA_CLASS>) = **{ class }**

FIRST(<ESTATUTOS>) = { } = **{ id, return, write, read, ++, --, if, while, do, for, ε }**

FIRST(<TIPO_DE_DATO>) = { } = **{ int, float, char, string, bool, void }**

FIRST(<VALOR_CONSTANTE>) = { } = **{ cteentera, ctereal, ctenotacion, ctecaracter, ctedtring }**

FIRST(<PARAMETROS>) = { } = **{ id, ε }**
FIRST(<PARAMETROS2>)= { } = **{ id, ε }**
FIRST(<PARAMETROS2A>)= { } = **{ , ε }**

FIRST(<EST_ID>)= { } = **{ id }**
FIRST(<EST_ID'>)= { } = **{ =, ++, -- }**

FIRST(<EST_RETURN>)= { } = **{ return }**

FIRST(<EST_WRITE>)= { } = **{ write }**
FIRST(<EST_READ>)= { } = **{ read }**

FIRST(<POSTINCUNARIO>)= { } = **{ ++, -- }**
FIRST(<PREINCUNARIO>)= { } = **{ ++, -- }**

FIRST(<EST_IF>)= { } = **{ if }**
FIRST(<EST_IF2>)= { } = **{ elseif, ε }**
FIRST(<EST_IF3>)= { } = **{ else, ε }**

FIRST(<EST_WHILE>)= { } = **{ while }**
FIRST(<EST_DO>)= { } = **{ do }**

FIRST(<EST_FOR>)= { } = **{ for }**

FIRST(<EXPR>)= { } = **{ id, cteentera, ctereal, ctenotacion, ctecaracter, ctedtring, (, ! }**
FIRST(<EXPR'>)= { } = **{ ||, ε }**

FIRST(<EXPR2>)= { } = **{ !, id, cteentera, ctereal, ctenotacion, ctecaracter, ctedtring, (}**
FIRST(<EXPR2'>)= { } = **{ &&, ε }**

FIRST(<EXPR3>)= { } = **{ !, id, cteentera, ctereal, ctenotacion, ctecaracter, ctedtring, (}**

FIRST(<EXPR4>)= { } = **{ id, cteentera, ctereal, ctenotacion, ctecaracter, ctedtring, (}**
FIRST(<EXPR4_2>)= { } = **{ ==, !=, <, <=, >, >=, ε }**

FIRST(<EXPR5>)= { } = **{ id, cteentera, ctereal, ctenotacion, ctecaracter, ctedtring, (}**
FIRST(<EXPR5'>) = { } = **{ +, -, ε }**

Analizador Sintactico

FIRST(<OPREL>) = { id, cteentera, ctereal, ctenotacion, ctecaracter, ctestring, (}

FIRST(<FACT>)= { } = **{ id, cteentera, ctereal, ctenotacion, ctecaracter, ctedtring, (}**

FIRST(<TERM>) = { } = **{ id, cteentera, ctereal, ctenotacion, ctecaracter, ctedtring, (}**
FIRST(<TERM'>)= { } = **{ * / % , ε }**

FIRST(<LLAMADA_F>)= { } = **{ (ε }**
FIRST(<LLAMADA_F2>)= { } = **{ id ε }**
FIRST(<LLAMADA_F2A>) = **{ , ε }**

Follows:

FOLLOW(<PROGRAM>) = **{ $ }**

FOLLOW(<DECLARE_LIBRARY>) = **{ def, const, function, class }**

FOLLOW(<PROGRAM2>) = **{ class }**

FOLLOW(<DECLARE_VAR>) = **{ def, const, function, class }**
FOLLOW(<DECLARE_VAR2>) = **{ def, const, function, class }**
FOLLOW(<DECLARE_VAR2A>) = **{ of }**

FOLLOW(<DECLARACONST>) = **{ def, const, function, class }**

FOLLOW(<DECLARA_FUNCTION>) = **{ def, const, function, class }**

FOLLOW(<DECLARA_CLASS>) = **{ $ }**

FOLLOW(<ESTATUTOS>) = **{ endif, elseif, else, endwhile, dowhile, enddo, endfor, class }**

FOLLOW(<EST_ID>) = **{ ; }**
FOLLOW(<EST_ID'>) = **{ ; }**

FOLLOW(<TIPO_DE_DATO>) = **{ ;) }**

FOLLOW(<VALOR_CONSTANTE>) = **{ ; }**

FOLLOW(<PARAMETROS>) = **{) }**
FOLLOW(<PARAMETROS2>) = **{ =) }**
FOLLOW(<PARAMETROS2A>) = **{) }**
FOLLOW(<EST_RETURN>) = **{ ; }**

FOLLOW(<EST_WRITE>) = **{ ; }**

FOLLOW(<EST_READ>) = **{ ; }**
FOLLOW(<EST_READ2>) = **{) }**

FOLLOW(<POSTINCUNARIO>) = **{ ; }**
FOLLOW(<POSTINCUNARIO'>) = **{ ; }**
FOLLOW(<PREINCUNARIO>) = **{ ; }**

FOLLOW(<EST_IF>) = **{ endif, elseif, else, endwhile, do while, enddo, endfor, class }**
FOLLOW(<EST_IF2>) = **{ else, endif }**
FOLLOW(<EST_IF3>) = **{ endif }**
FOLLOW(<EST_WHILE>) = **{ endif, elseif, else, endwhile, dowhile, enddo, endfor, class }**
FOLLOW(<EST_DO>) = **{ endif, elseif, else, endwhile, dowhile, enddo, endfor, class }**
FOLLOW(<EST_FOR>) = **{ endif, elseif, else, endwhile, dowhile, enddo, endfor, class }**

FOLLOW(<EXPR>) = **{) ; , to }**
FOLLOW(<EXPR'>) = **{) ; , to }**
FOLLOW(<EXPR2>) = **{ ||) ; , to }**
FOLLOW(<EXPR2'>) = **{ ||) ; , }**

Analizador Sintactico

FOLLOW(<EXPR3>) = { &&) ; , }
FOLLOW(<EXPR4>) = { &&) ; , }
FOLLOW(<EXPR4_2>) = { &&) ; , }
FOLLOW(<EXPR5>) = { == != < <= > >= && ||) ; , }
FOLLOW(<EXPR5'>) = { == != < <= > >= && ||) ; , }
FOLLOW(<OPREL>) = { id cteentera ctereal ctenotacion ctecaracter ctedtring to (!}

FOLLOW(<TERM>) = { + - == != < <= > >= && ||) ; ,) to }
FOLLOW(<TERM'>) = { + - == != < <= > >= && ||) ; ,) to }
FOLLOW(<FACT>) = { * ** / % + - == != < <= > >= && ||) ; ,) to}

Matriz Predictiva:

https://docs.google.com/spreadsheets/d/18KA5T4D--srFuljZlJSX6Hp1J8hHWEigv146Ar1ov1M/edit?gid=0#gid=0

Matriz Producciones:

Analizador Sintactico

1	<PROGRAM>	2	4	8								
2	<DECLARE_LIBRARY>	135	136	137	138	2	139					
3	<DECLARE_LIBRARY>	-100										
4	<PROGRAM2>	9	4									
5	<PROGRAM2>	14	4									
6	<PROGRAM2>	16	4									
7	<PROGRAM2>	-100										
8	<DECLARA_CLASS>	140	17	141								
9	<DECLARAVAR>	142	11	143	30	147						
10	<DECLARAVAR>	-100										
11	<DECLARAVAR2>	136	12									
12	<DECLARAVAR2A>	-100										
13	<DECLARAVAR2A>	145	136	12								
14	<DECLARACONST>	145	136	146	36	147						
15	<DECLARACONST>	-100										
16	<DECLARA_FUNCTION>	148	136	146	30	149	41	150	9	17	151	
17	<ESTATUTOS>	27	147	17								
18	<ESTATUTOS>	47	147	17								
19	<ESTATUTOS>	48	147	17								
20	<ESTATUTOS>	51	147	17								
21	<ESTATUTOS>	57	147	17								
22	<ESTATUTOS>	59	17									
23	<ESTATUTOS>	64	17									
24	<ESTATUTOS>	65	17									
25	<ESTATUTOS>	66	17									
26	<ESTATUTOS>	-100										
27	<EST_ID>	136	28									
28	<EST_ID'>	146	67									
29	<EST_ID'>	54										
30	<TIPO_DE_DATO>	152										
31	<TIPO_DE_DATO>	153										
32	<TIPO_DE_DATO>	154										
33	<TIPO_DE_DATO>	155										
34	<TIPO_DE_DATO>	156										
35	<TIPO_DE_DATO>	157										
36	<VALOR_CONSTANTE>	158										
37	<VALOR_CONSTANTE>	159										
38	<VALOR_CONSTANTE>	160										
39	<VALOR_CONSTANTE>	161										
40	<VALOR_CONSTANTE>	162										
41	<PARAMETROS>	44	146	30	144	41						
42	<PARAMETROS>	44	146	30								
43	<PARAMETROS>	-100										
44	<PARAMETROS2>	136	46									
45	<PARAMETROS2>	-100										
46	<PARAMETROS2A>	144	136	46								
47	<EST_RETURN>	163	67									
48	<EST_WRITE>	164	149	67	49	150						
49	<EST_WRITE2>	144	67	49								
50	<EST_WRITE2>	-100										
51	<EST_READ>	165	149	136	52	150						
52	<EST_READ2>	144	136	52								
53	<EST_READ2>	-100										
54	<POSTINCUNARIO>	55										
55	<POSTINCUNARIO'>	166										
56	<POSTINCUNARIO'>	167										
57	<PREINCUNARIO>	166	136									
58	<PREINCUNARIO>	167	136									

Analizador Sintactico

59	<EST_IF>	168	149	67	150	17	60	62	169	
60	<EST IF2>	170	149	67	150	17	60			
61	<EST IF2>	-100								
62	<EST IF3>	171	17							
63	<EST IF3>	-100								
64	<EST_WHILE>	172	149	67	150	17	173			
65	<EST_DO>	174	17	175	149	67	150	176		
66	<EST_FOR>	177	136	149	67	178	67	150	17	179
67	<EXPR>	70	68							
68	<EXPR'>	180	70	68						
69	<EXPR'>	-100								
70	<EXPR2>	73	71							
71	<EXPR2'>	181	73	71						
72	<EXPR2'>	-100								
73	<EXPR3>	182	75							
74	<EXPR3>	75								
75	<EXPR4>	78	76							
76	<EXPR4_2>	82	78							
77	<EXPR4_2>	-100								
78	<EXPR5>	87	79							
79	<EXPR5'>	183	78							
80	<EXPR5'>	184	78							
81	<EXPR5'>	-100								
82	<OPREL>	185								
83	<OPREL>	186								
84	<OPREL>	187								
85	<OPREL>	188								
86	<OPREL>	189								
87	<OPREL>	190								
88	<TERM>	94	89							
89	<TERM'>	191	94	89						
90	<TERM'>	192	94	89						
91	<TERM'>	193	94	89						
92	<TERM'>	194	94	89						
93	<TERM'>	-100								
94	<FACT>	136	97							
95	<FACT>	36								
96	<FACT>	149	67	150						
97	<LLAMADA_F>	149	99	150						
98	<LLAMADA_F>	-100								
99	<LLAMADA_F2>	136	101							
100	<LLAMADA_F2>	-100								
101	<LLAMADA_F2A>	144	136	101						
102	<LLAMADA_F2A>	-100								

Analizador Sintactico

Enumeraciones de la matriz de producciones:

135- include
136- id
137- .
138- lib
139- endlib
140- class
141- endaclass
142- def
143- of
144- ,
145- const
146- =
147- ;
148- function
149- (
150-)
151- endfunction
152- int
153- float
154- char
155- string
156- bool
157- void
158- cteentera
159- ctereal
160- ctenotacion
161- ctecaracter
162- ctedtring
163- return
164- write
165- read
166- ++
167- --
168- if
169- endif
170- elseif
171- else
172- while
173- endwhile
174- do
175- dowhile
176- enddo
177- for
178- to
179- endfor
180- ||
181- &&
182- !
183- +
184- -
185- ==
186- !=
187- <
188- <=
189- >
190- >=
191- *
192- /
193- %
194- **

Analizador Sintactico

include math.lib;