

Pontifícia Universidade Católica de Minas Gerais  
Instituto de Ciências Exatas e Informática  
Otimização de Sistemas

Documentação:  
Compilador linguagem L

Iago Augusto Coelho Morgado  
Homenique Vieira Martins  
Guilherme Cósso Lima Pimenta

## Tabela de Token e Lexemas

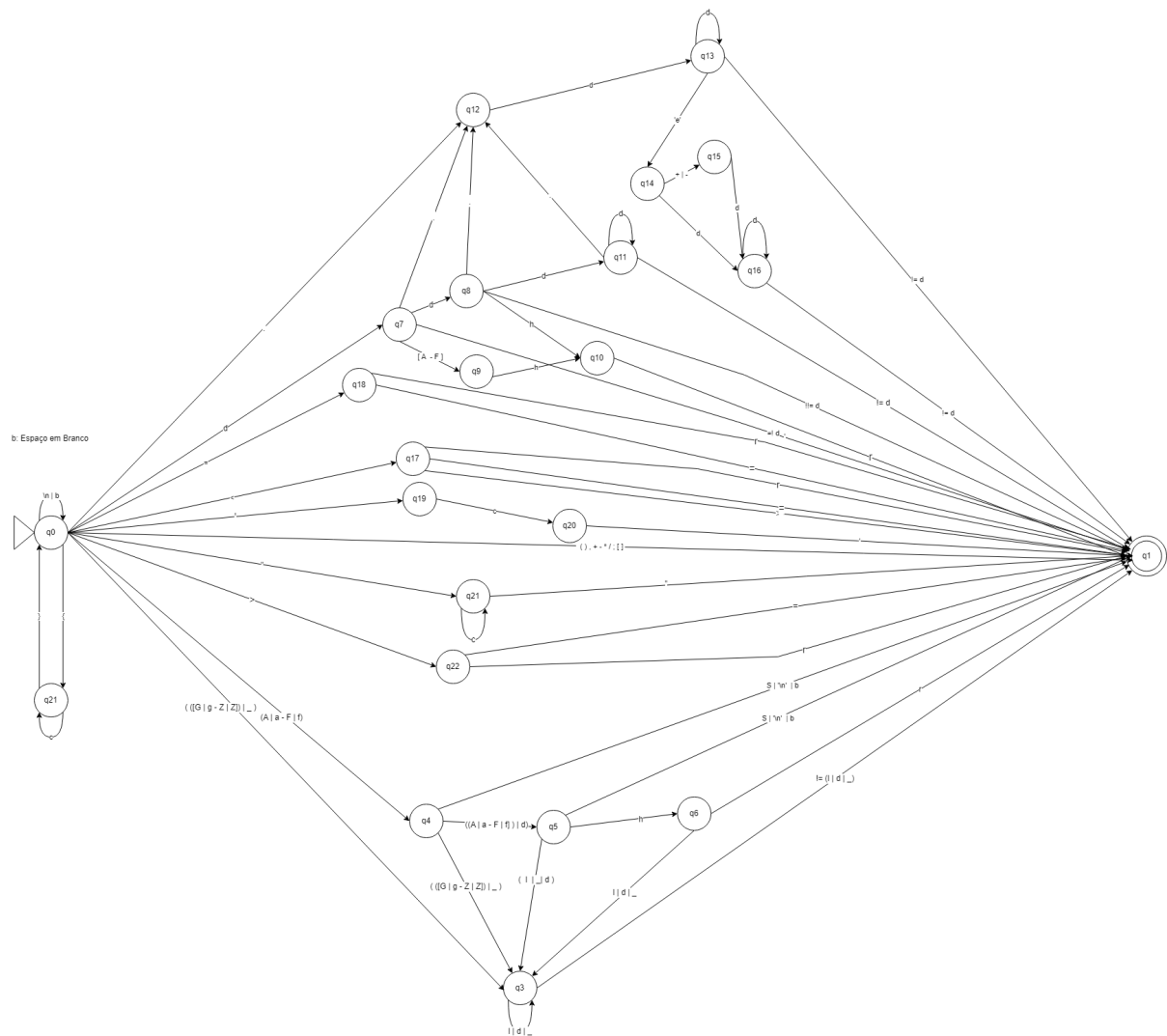
---

Nº	Token	Lexema
1	final	'f' 'i' 'n' 'a' 'l'
2	integer	'i' 'n' 't' 'e' 'g' 'e' 'r'
3	char	'c' 'h' 'a' 'r'
4	real	'r' 'e' 'a' 'l'
5	string	's' 't' 'r' 'i' 'n' 'g'
6	boolean	'b' 'o' 'o' 'l' 'e' 'a' 'n'
7	for	'f' 'o' 'r'
8	if	'i' 'f'
9	else	'e' 'l' 's' 'e'
10	and	'a' 'n' 'd'
11	or	'o' 'r'
12	not	'n' 'o' 't'
13	==	'=' '='
14	=	'='
15	>	'>'
16	>=	'>' '='
17	<	'<'
18	<=	'<' '='
19	<>	'<' '>'
20	(	'('
21	)	')'
22	[	'['
23	]	']'
24	,	','
25	+	'+'

26	-	'_'
27	div	( 'd' 'i' 'v'   '/' )
28	*	'*'
29	mod	'm' 'o' 'd'
30	const	( ( <b>HEX</b>   <b>D</b> ) ( <b>HEX</b>   <b>D</b> ) 'h' ) ∪ ( ( <b>D</b> *. <b>D</b> + [e [- +] <b>D</b> +])   <b>D</b> + ) ∪ ( ' [ <b>C</b> ] ' ) ∪ ( " <b>C</b> * ")
31	id	( <b>L</b>   _ )( <b>L</b>   <b>D</b>   _ )*
32	;	'.'
33	:	'.'
34	false	'f' 'a' 'l' 's' 'e'
35	true	't' 'r' 'u' 'e'
36	begin	'b' 'e' 'g' 'i' 'n'
37	end	'e' 'n' 'd'
38	write	'w' 'r' 'i' 't' 'e'
49	writeln	'w' 'r' 'i' 't' 'e' 'l' 'n'
40	readln	'r' 'e' 'a' 'd' 'l' 'n'
41	fim_arquivo	'eof'

Legenda	
<b>C</b>	qualquer caractere aceito pela linguagem (letra, dígito , espaço, sublinhado, etc)
<b>L</b>	qualquer letra do alfabeto
<b>D</b>	qualquer dígito de 0 a 9
<b>HEX</b>	qualquer caractere do alfabeto de a até f (a,b,c,d,e,f), característico de números hexadecimais

# Analizador Léxico



# Gramática

---

**S** -> { ( **D** | **Cmd** ) } **fim\_arquivo**  
//Declarações  
**D** -> **D1 C** { ,**C** } ; | **D2**  
**C** -> **id** [ ( = ( [ - ] **const** | **true** | **false** ) ) ] | ( [ ( **id** | **const** ) ] )  
**D1** -> ( **char** | **integer** | **real** | **boolean** | **string** )  
**D2** -> **final id** = [ - ] **const** ;  
//Comandos  
**Cmd** -> ( ( [ **A** | **L** | **E** ] ; ) | ( **R** | **T** ) )  
**Cmd1** -> [ ( **A** | **R** | **T** | **L** | **E** ) ]  
**A** -> **id** [ [ **Exp** ] ] = **Exp**  
**R** -> **for** ( **R1** ; **Exp** ; **R1** ) **T1**  
**R1** -> **Cmd1** { ,**Cmd1** }  
**T** -> **if** ( **Exp** ) **T1** [ **else T1** ]  
**T1** -> ( **Cmd** | **begin** { **Cmd** } **end** )  
**L** -> **readln** ( **id** )  
**E** -> ( **write** | **writeln** ) ( **E1** )  
**E1** -> **Exp** { , **Exp** }  
//Expressão  
**Exp** -> **Exp1** { ( == | < | <= | > | >= | <> ) **Exp1** }  
**Exp1** -> [ - ] **Exp2** { ( + | - | **or** ) **Exp2** }  
**Exp2** -> **Exp3** { ( \* | **mod** | ( **div** | / ) | **and** ) **Exp3** }  
**Exp3** -> **Exp4** | **not** **Exp4**  
**Exp4** -> **Exp5** | **real**( **Exp5** ) | **integer**( **Exp5** )  
**Exp5** -> **const** | **true** | **false** | [ **id** [ ( [ **id** | **const** ) ] ] | ( **Exp** )

Legenda	
Azul	Símbolo Terminal
Vermelho	Símbolo não terminal