

Extended Backus-Naur-Form

start ::= begin, cmd_var, cmd, end

begin ::= "begin"

end ::= "end"

cmd_var ::= (tipo_var, var, ponto_virgula) [cmd_var]

tipo_var ::= "int" | "flut"

var ::= nome_var [atribuicao] [virgula, var]

ponto_virgula ::= ";"

cmd ::= cmd_se | cmd_enquanto | (atribuicao, ponto_virgula) [cmd]

nome_var ::= {letra}

atribuicao ::= "=" valor_var

cmd_se ::= se , abre_parenteses, condicao, fecha_parenteses, abre_chaves, cmd, fecha_chaves

se ::= "se"

cmd_enquanto ::= enquanto, abre_parenteses , condicao, fecha_parenteses, abre_chaves, cmd, fecha_chaves

enquanto ::= "enquanto"

abre_parenteses ::= "("

fecha_parenteses ::= ")"

abre_chaves ::= "{"

fecha_chaves ::= "}"

valor_var ::= [-] numero | nome_var | (numero, operador_aritmetico , numero) | (nome_var , operador_aritmetico , numero)

condicao ::= (nome_var, operador_relacional, nome_var) | (nome_var, operador_relacional, numero)

numero ::= {digitos}+ | ({digitos}+, ".", {digitos}+)

digitos ::= 0-9

operador_relacional ::= "<" | ">" | "==" | "!=" | ">=" | "<="

operador_aritmetico ::= "+", "-", "*", "/"

Letra ::= letra_minuscula | letra_maiuscula

letra_minuscula ::= a-z

letra_maiuscula ::= A-Z

Expressão	First	Follow
start	first(begin)	{'{'}
begin	{'begin'}	first(cmd_var)
end	{'end'}	{'{'}
cmd_var	first(tipo_var)	first(cmd)
var	first(nome_var)	first(ponto_virgula)
ponto_virgula	{';'}	first(cmd_var) U first(cmd)
cmd	{'se', 'enquanto'} U first(atribuicao)	first(end) U first(fecha_chave)
tipo_var	{int, flut, frase}	first(nome_var)
nome_var	first(letra)	first(atribuicao) U ',' U first(ponto_virgula) U first(operador_relacion al) U first(operador_aritmetic o) U first(fecha_parenteses)
atribuicao	'='	first(ponto_virgula) U ','
cmd_se	{'se'}	follow(cmd)
se	{'se'}	{'{'}
cmd_enquanto	{'enquanto'}	follow(cmd)
enquanto	{'enquanto'}	{'{'}
abre_parenteses	{'{'}	first(condicao)
fecha_parenteses	{'}'}	first(abre_chaves)
abre_chaves	{'{'}	first(cmd)
fecha_chaves	{'}'}	first(cmd) U follow(cmd)

valor_var	first(numero)	',' U first(ponto_virgula)
condicao	first(nome_var)	first(fecha_parenteses)
numero	first(digitos)	first(operador_aritmetico) U ',' U ponto_virgula
digitos	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}	first(digitos)
operador_relacional	{< , > , == , != , >= , <=}	first(nome_var) U first(numero)
operador_aritmetico	{+, -, *, /}	
letra	first(letra_minuscula) U first(letra_maiuscula)	first(letra) U ',' U ponto_virgula
letra_minuscula	{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z}	follow(letra)
letra_maiuscula	{A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z}	follow(letra)