

Extended Backus-Naur Form

linguagem = **"begin"** comandos* **"end"**

comandos = variaveis | atribuicao | condicao | repeticao

variaveis = tipo id ["=" valor] ("," id ["=" valor])* ";"

atribuicao = id "=" valor ";"

id = letra (letra | digito | "_")*

valor = expressao_mat | cadeia

expressao_mat = "(" expr_mat ")" | exp_mat

exp_mat = ["-"] (id | numero) [operador_mat expressao_mat]

operador_mat = "+" | "-" | "*" | "/" | "%" | "^"

numero = digito+ ["." digito+] ["E" ["+" | "-") digito+]

cadeia = "\"" (letra | digito | simbolo)* "\""

tipo = **"number"** | **"string"**

letra = "a" | "b" | "c" | "d" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z" | "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"

digito = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

simbolo = "@" | "!" | "?" | "#" | "&" | "%" | "\$" | "_"

condicao = **"if"** "(" verifica ")" entre [**"else"** entre]

entre = "{" comandos* "}"

verifica = ["!"] pergunta

repeticao = for | while

for = **"for"** "(" id "=" expressao_mat ";" verifica ";" passo ")" entre

passo = id (incremento1 | incremento2)

incremento1 = ("+=" | "-=" | "*=" | "/=") expressao_mat

incremento2 = "++" | "--"

pergunta = valor operador_rel valor

operador_rel = ">" | "<" | "!=" | ">=" | "<=" | "=="

while = **"while"** "(" verifica ")" entre

Gramática Regular

LINGUAGEM = "início" COMANDOS "final"

COMANDOS = ϵ | VARIAVEIS COMANDOS | ATRIBUICAO COMANDOS | CONDICAO COMANDOS | REPETICAO COMANDOS

VARIAVEIS = TIPO "id" IGUAL_VALOR CONTINUACAO ","

IGUAL_VALOR = ϵ | "=" VALOR

CONTINUACAO = ϵ | "," "id" IGUAL_VALOR CONTINUACAO

ATRIBUICAO = "id" "=" VALOR ","

VALOR = EXPRESSAO_MAT | "cadeia"

EXPRESSAO_MAT = "(" EXP_MAT ")" | EXP_MAT

EXP_MAT = MENOS_OP VAR CONTA

CONTA = ϵ | OPERADOR_MAT EXPRESSAO_MAT

VAR = "id" | "numero"

MENOS_OP = ϵ | "-"

OPERADOR_MAT = "+" | "-" | "*" | "/" | "%" | "^"

CONDICAO = "se" "(" VERIFICA ")" ENTRE ELSE

ELSE = ϵ | "senão" ENTRE

ENTRE = "{" COMANDOS "}"

VERIFICA = NOT PERGUNTA

NOT = ϵ | "!"

REPETICAO = FOR | WHILE

FOR = "for" "(" "id" "=" EXPRESSAO_MAT "," VERIFICA "," PASSO ")" ENTRE

PASSO = "id" INCREMENTO

INCREMENTO = INCREMENTO1 | INCREMENTO2

INCREMENTO1 = "+=" EXPRESSAO_MAT | "-=" EXPRESSAO_MAT | "*=" EXPRESSAO_MAT | "/=" EXPRESSAO_MAT

INCREMENTO2 = "++" | "--"

PERGUNTA = VALOR OPERADOR_REL VALOR

OPERADOR_REL = ">" | "<" | "!=" | ">=" | "<=" | "=="

WHILE = "while" "(" VERIFICA ")" ENTRE

TIPO = “number” | “string”

Observações:

“id” representa as letras e suas sequências válidas

“numero” representa os números e suas sequências válidas

Definição de First e Follow da Gramática

| Não Terminal | First (P) | Follow (S) | Regra |
|---------------|---|--|---|
| LINGUAGEM | inicio | \$ | |
| VARIAVEIS | number, string | final, } | P(COMANDOS) + S(COMANDOS) |
| COMANDOS | ϵ , number, string, id, if, for, while | final, } | final + } + S(COMANDOS) |
| IGUAL_VALOR | ϵ , = | , , ; | P(CONTINUACAO) + S(CONTINUACAO) + ; |
| CONTINUACAO | ϵ , , | ; | ; + S(CONTINUACAO) |
| ATRIBUICAO | id | numero, string, id, if, for, while, final, } | P(COMANDOS) + S(COMANDOS) |
| VALOR | -, (, cadeia, id, numero | , , ;, >, <, !=, >=, <=, ==, ,) | S(IGUAL_VALOR) + ; + P(OPERADOR_REL) + S(PERGUNTA) |
| EXPRESSAO_MAT | -, (, id, numero | , , ;, >, <, !=, >=, <=, ==, ,) | S(VALOR) + ; + S(CONTA) + S(INCREMENTO1) |
| EXP_MAT | -, id, numero | , , ;, >, <, !=, >=, <=, ==, ,) |) + S(EXPRESSAO_MAT) |
| CONTA | ϵ , +, -, *, /, %, ^ | , , ;, >, <, !=, >=, <=, ==, ,) | S(EXP_MAT) |
| VAR | id, numero | +, -, *, /, %, ^, , , ;, >, <, !=, >=, <=, ==, ,) | P(CONTA) + S(EXP_MAT) |
| MENOS_OP | ϵ , - | id, numero | P(VAR) |
| OPERADOR_MAT | +, -, *, /, %, ^ | -, (, id, numero | P(EXPRESSAO_MAT) |
| CONDICAO | if | numero, string, id, if, for, while, final, } | P(COMANDOS) + S(COMANDOS) |
| ENTRE | { | else, numero, string, id, if, for, while, final, } | P(ELSE) + S(CONDICAO) + S(ELSE) + S(FOR) + S(WHILE) |
| ELSE | ϵ , else | numero, string, id, if, for, while, final, } | S(CONDICAO) |
| VERIFICA | -, (, cadeia, id, numero, ! | , , ; |) + ; |
| NOT | ϵ , ! | -, (, cadeia, id, numero | P(PERGUNTA) |
| REPETICAO | for, while | numero, string, id, if, for, while, final, } | P(COMANDOS) + S(COMANDOS) |
| FOR | for | numero, string, id, if, for, while, final, } | S(REPETICAO) |
| WHILE | while | numero, string, id, if, for, while, final, } | S(REPETICAO) |
| PASSO | id |) |) |
| INCREMENTO | +=, -=, *=, /=, ++, -- |) | S(PASSO) |
| INCREMENTO1 | +=, -=, *=, /= |) | S(INCREMENTO) |
| INCREMENTO2 | ++, -- |) | S(INCREMENTO) |

| | | | |
|--------------|-----------------------------|-----------------------------|-------------|
| PERGUNTA | - , (, cadeia , id, numero |) , ; | S(VERIFICA) |
| OPERADOR_REL | > , < , != , >= , <= , == | - , (, cadeia , id, numero | P(VALOR) |
| TIPO | number, string | id | id |