**Dicionário**

for = Em\_tempo V

if = nu\_jeito V

else = Causo V

else if = Ques\_sera V

Do = É\_mió V

while = Logo\_ai V

function = Encasquetá V

Break = po\_para V

Continue = peraí V

End / Exit = Cascar\_fora V

int = Interim V

char = Enjoado V

string = to\_ti\_falano V

double = Enrabichado V

Long = Logali V

float = Cadin V

Boolean = Meia\_boca V

printf = Dediprosa V

scanf = Panhá V

Array = Encantoá V

List = Trenheira V

False = Bagaça V

True = Di\_vera V

Return = Arreda V

Enum = Uai V

Void = Ontocó V

Static = Paradeza V

Switch = Qualé V

Case = Qui\_nem V

Default = Murrinha V

Tuple = pelejanu V

**BNF**

<DefinirFuncao> ::= <TipoDeDados> 'encasqueta' id '(' <ListaParametro> ')' '{' <ListaComando> '}'

<ListaParametro> ::= <Parametro><ListaParametro2>

| <ListaParametro2> ::= ',' <Parametro><ListaParametro2>

| <Paramentro> ::= <TipoDeDados> ' 'id

<Digito> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<Tipo\_De\_Dados> ::= 'inteirim' /\* int \*/

| 'cadin' /\* float \*/

| to\_ti\_falano /\* string \*/

| 'ontoco' /\* void \*/

| ‘enjoado’ / \* char \*/

| ‘enrabichado’ /\* double \*/

| ‘logali ’ /\* long\*/

| ‘encantoa’ /\* array \*/

| ‘trenheira’ /\* list \*/

| ‘meia\_boca’ /\* boolean \*/

| ‘uai’ /\* enum \*/

| ‘pelejanu’ /\* tuple\*/

| ‘paradeza’ /\* static \*/

<ListaComando> ::= | <Comando><ListaComando>

<Comando> ::= <Declaracao>

| <Variavel>

| <em\_tempo> /\* for \*/

| <nu\_jeito> /\* if \*/

| <causo> /\* else \*/

| <ques\_sera /\*else if\*/

| <arreda> /\* return \*/

| <panha> /\* input \*/

| <dediprosa> /\* output \*/

| <Chamada>';'

| '{'<ListaComando>'}'

<Comando\_quale> ::= | <po\_para> /\*break \*/

| <perai> /\*continue\*/

<Declaracao> ::= <TipoDeDados> <ListaId> '=' ';'

<ListaId> ::= id <ListaId2>

<ListaId2> ::= ','<ListaId2> |

<Atributo> ::= id '=' <ExpressaoAritimetica> ';' //

<ExpressaoAritimetica> ::= <Termo><ExpressaoAritimetica2>

<ExpressaoAritimetica2> ::= <Operador\_aritimetico><ExpressaoAritimetica> |

<Termo> ::= <Fator><Termo2>

<Termo2> ::= <Operador\_aritimetico<Termo> |

<Fator> ::= <Operando>

| <Chamada>

| '('<ExpressaoAritimetica>')'

<Operando> ::= id

| cli /\* constante literal integer \*/

| clf /\* constante literal float \*/

| cls /\* constante literal string \*/

| clc /\* constante literal char \*/

| cld /\* constante literal double \*/

| cla /\* constante literal array \*/

| cll /\* constante literal list \*/

| clb /\* constante literal boolean \*/

| cle /\* constante literal enum \*/

| clt /\* constante literal tuple \*/

<Operador\_aritimetico> ::= '+'

| '-'

| ‘\*’

| '/'

<Operador\_logico> ::= ‘&&’

| ‘||’

| ‘!’

<Operador\_relacional> ::= '<'

| '>'

| '>='

| '<='

| '=='

| '!=’

<Operador\_atribuicao> ::= ‘+=’

| ‘-=’

| ‘\*=’

| ‘/=’

<ExpressaoRelacional> ::=

<ExpressaoAritimetica><Operador\_relacional><ExpressaoAritimetica>

<Pre\_Incrementacao> ::= ‘++’ id

<Pre\_Decrementacao> ::= ‘--’ id

<Pos\_Incrementacao> ::= id ‘++’

<Pos\_Decrementacao> ::= id ‘--’

<Expressao\_Declaracao> ::= <ExpressaoRelacional>

| <Pre\_Incrementacao>

| <Pre\_Decrementacao>

| <Pos\_Incrementacao>

| <Pos\_Decrementacao>

<nu\_jeito> ::= 'nu\_jeito' '('<ExpressaoRelacional>')' <Comando> <causo> /\* if\*/

<ques\_sera> ::= ‘ques\_sera’ '(' <ExpressaoRelacional> ')' <Comando>

<causo> ::= 'causo' <Comando> | /\* else \*/

<arreda> ::= 'arreda' <Fator> ';' /\* return \*/

<panha> ::= 'panha' '(' id ')' ';' /\* input \*/

<dediprosa> ::= 'dediprosa' '('<Operando>')'';' /\* output \*/

<po\_para> ::= ‘po\_para’ ‘;’ /\* break\*/

<perai> ::= ‘perai’ ‘;’ /\* continue \*/

<Chamada> ::= 'id' '('<ListaArgumento>')' ';'

<ListArgumento> ::= <Operando><ListArgumento2> |

<ListArgumento2> ::= ','<Operando><ListArgumento2>

<meia\_boca> ::= di\_vera /\* true \*/

| bagaca /\* false \*/

/\*switch, case, continue, break \*/

<quale> ::= ‘quale’ ‘(‘ id ‘)’ <bloco\_quale>

<bloco\_quale> ::= ‘{‘ <grupo\_bloco\_quale>? <quale\_nome>? ‘}’

<grupo\_blocos\_quale> ::= <grupo\_bloco\_quale>

| <grupo\_bloco\_quale> <grupo\_bloco\_quale>

| <quale\_nome><grupo\_bloco\_quale>

<quale\_nome> ::= <quale\_nome>

| <quale\_nome> <quale\_nome>

| ‘qui\_nem ’ id ‘:’ <ListaComando>

| ‘qui\_nem ’ id ‘:’ <comando\_quale>

| ‘murrinha:’ <ListaComando>

/\* for \*

<em\_tempo> ::= em\_tempo ‘(‘ <em\_tempo\_incio> ‘;’ <ExpressaoRelacional> ‘;’ <em\_tempo\_atualizar> ‘)’ ‘{’ <ListaComando> ‘}’

<em\_tempo\_linha> ::= em\_tempo ‘(‘<em\_tempo\_incio> ‘;’ <ExpressaoRelacional> ‘;’ <em\_tempo\_atualizar> ‘)’ <Comando> ‘;’

<em\_tempo\_incio> ::= <Lista\_Expressao\_Declaracao>

| <Declaracao>

<em\_tempo\_atualizar> ::= <Lista\_Expressao\_Declaracao>

<Lista\_Expressao\_Declaracao> ::= <Expressao\_Declaracao>

| <Lista\_Expressao\_Declaracao> , <Expressao\_Declaracao>

/\* while, do while \*/

<logo\_ai> ::= ‘logo\_ai’ ‘(‘ <ExpressaoRelacional> ')’ '{' <ListaComando> '}'

<logo\_ai\_linha> ::= ‘logo\_ai’ ‘(‘ <ExpressaoRelacional> ‘)’ <Comando> ‘;’

<e\_mio> ::= ‘{‘ <ListaComando> ‘}’ ‘logo\_ai’ ‘(‘ <ExpressaoRelacional> ‘)’ ;

/\* Array \*/

<encontoa> ::= <tipo\_de\_dados>‘[‘ ‘]’ id ‘=’ ‘{’ id ‘,’ <encontoa\_expressao> ‘}’ ‘;’

<encontoa\_expressao> ::= id ‘,’ <encontoa\_expressao> ‘;’

/\* Tuple \*/

<pelejanu> ::= id ‘=’ '(' id ',' <pelejanu\_expressao> ')' ‘;’

<pelejanu\_expressao> ::= id ‘,’ <pelejanu\_expressao>

/\* list \*/

<trenheira> ::= id ‘=’ '[' id ',' <trenheira \_expressao> ']' ‘;’

<trenheira \_expressao> ::= id ‘,’ <trenheira \_expressao>

/\* string, char \*/

<enjoado> ::= ' <enjoado\_unico> '

| ' <sequencia\_fuga> '

<enjoado\_unico> ::= <input\_enjoado> except ' and \

<to\_ti\_falano> ::= " < to\_ti\_falano\_caracteres>"

<to\_ti\_falano\_caracteres> ::= <to\_ti\_falano\_caracter>

| <tto\_ti\_falano\_caracteres> <o\_ti\_falano\_caracter>

<to\_ti\_falano\_caracter> ::= <input\_to\_ti\_falano> except " and \

| <escape\_to\_ti\_falano>

<Cascar\_fora> ::= ‘Cascar\_fora’ /\* end \*/

**Exemplo**

inteirim encasqueta Soma (inteirim a, inteirim b){

inteirim soma = a + b;

arreda soma;

}

ontoco encasqueta Dados (){

inteirim a = panha(“Digite um numero: ”);

inteirim b = panha(“Digite outro numero: ”);

inteirim resultado = soma(a, b);

dediprosa(resultado);

}